SOETO

Stołeczny Ośrodek
Elektronicznej
Techniki Obliczeniowej

INFORMATYKA mikrokomputerowa

INSTRUKCJA OBSŁUGI KOMPUTERA AMSTRAD CPC 6128

tom I

AMSTRAD

Warszawa 1986 r.

Książka zawiera tłumaczenie na język polski pierwszej części oryginalnej instrukcji obsługi komputera AMSTRAD CPC 6128. Podano w niej podstawowe zasady posługiwania się komputerem, zasady programowania w języku BASIC i podstawowe informacje o posługiwaniu się dyskami elastycznymi przy użyciu systemów operacyjnych AMSDOS i CP/M. Książka zawiera także pełny opis słów kluczowych stosowanej w CPC 6128 wersji języka BASIC.
Z książki mogą korzystać osoby nie mające żadnego przygotowania w tej dziedzinie.

Tłumaczenie przygotowane przez firmę wysyłkową
POLANGLIA LTD
jedyne autoryzowane przedstwicielstwo firmy AMSTRAD na Polskę.
58 St. Mary's Road
London W5 5EX
Wielka Brytania
tel.0-0441-840 1715
tlx 946581

INSTRUKCJA OBSŁUGI
KOMPUTERA AMSTRAD CPC 6128
tom I

Druk ZRiWDB Warszawa ul. Królewska 27

Format A-5 Nakład 3000 egz.   P-48

zam. nr 280 z 1986 XI 18   Ark. 16,25

WAŻNE

Musisz to przeczytać:

Uwagi o instalacji

1. Zawsze dołączaj przewód sieciowy do wtyczki o trzech końcówkach zgodnie z instrukcją zawartą w części 1 Kursu Podstawowego
2. Nigdy nie próbuj dołączać systemu do napięcia sieci innego niż 220-240 V, 50 Hz
3. Wewnątrz systemu nie ma części obsługiwanych przez użytkownika. Nie próbuj dostać się do wnętrza systemu. Zwróć się do przeszkolonego personelu serwisowego
4. Aby uniknąć zmęczenia oczu, monitor zawsze powinien być możliwie oddalony od klawiatury i pracować w odpowiednio oświetlonym pomieszczeniu. JASNOŚĆ monitora powinna być możliwie najniższa.
5. Komputer powinien być ustawiony na wprost monitora, ale jak najdalej od niego. Dla zapewnienia maksymalnej pewności działania, część komputera zawierająca napęd dysków NIE powinna być umieszczona przed monitorem, lecz na prawo od niego. Nie należy umieszczać komputera w sąsiedztwie źródeł zakłóceń elektrycznych
6. Zawsze trzymaj napęd dysków i same dyski z dala od pól magnetycznych
7. Jeżeli używasz systemu o 2 napędach, to przewód łączący drugi napęd winien być umieszczony z dala od przewodu sieciowego
8. Nie zakrywaj i nie blokuj otworów wentylacyjnych
9. Nie przechowuj systemu w środowisku szczególnie gorącym, zimnym lub o dużej zawartości kurzu

Uwagi o obsłudze

(Nie obawiaj się, jeżeli zostaniesz trochę oszołomiony występującym w tej części żargonem technicznym; ważność tych uwag stanie się jasna po przeczytaniu tego podręcznika).

1. Nigdy nie włączaj i nie wyłączaj systemu z dyskiem umieszczonym w napędzie. Takie działanie zniszczy Twój dysk i doprowadzi do utraty cennych programów lub danych

2. Zawsze wykonuj duplikaty (kopie) dysków zawierających cenne programy. Dotyczy to szczególnie dysków tworzących pakiet wzorcowych dysków systemowych CP/M, dołączonych do 6128. Jeżeli przypadkowo stracisz lub zniszczysz te dyski - to ich wymiana może okazać się bardzo kosztowna
3. Upewnij się, żeby nigdy nie zapisać przypadkowo dysków systemowych CP/M. Zapewnij, żeby otwory zabezpieczające przed zapisem były w tych dyskach zawsze otwarte
4. Jeżeli używasz systemu o 2 napędach, tzn. zakupiłeś stację dysków AMSTRAD FD1, to zawsze włączaj drugi napęd dyskowy przed włączeniem komputera
5. Nigdy nie dotykaj powierzchni dysku elastycznego wewnątrz obudowy ochronnej
6. Nigdy nie wyjmuj dysku w czasie, gdy odbywa się na nim zapis lub odczyt
7. Pamiętaj zawsze, że sformatowanie dysku wymaże jego poprzednią zawartość
8. Wewnętrzny dysk zajmuje mały obszar pamięci, który używany był czasami przez zawodowych programistów - autorów oprogramowania na kasetach w modelu AMSTRAD CPC 464. Kasety te nie będą działać właściwie na 6128 z dołączonym magnetofonem. Jeżeli masz jakiekolwiek wątpliwości dotyczące kompatibilności oprogramowania na kasetach - skontaktuj się z AMSOFT w Brentwood (0277) 230222. Pamiętaj jednak, że większość podstawowego oprogramowania AMSOFT dostępna jest na dyskach dla 6128
9. Umowa licencyjna na Twoje dyski systemowe CP/M, (które są oznaczone numerem seryjnym w sposób elektroniczny) zezwala na używanie ich tylko w pojedynczym systemie komputerowym. W szczególności oznacza to, że nie wolno Ci przekazywać innej osobie dysku CP/M z zakodowanym TWOIM numerem seryjnym kopii systemu operacyjnego.

ROZDZIAŁ 1
KURS PODSTAWOWY

Część 1: Połączenie komputera

Komputer CPC 6128 może być połączony z jednym z następujących urządzeń:

1. Monitor zielony AMSTRAD GT65
2. Monitor kolorowy AMSTRAD CTM 664
3. Modulator/zasilacz AMSTRAD MP2 i odbiornik telewizji kolorowej z pasmem UHF

Dostosowanie wtyczki sieciowej

CPC 6128 zasilany jest z sieci 220-240 V, 50 Hz. Do przewodu sieciowego, wychodzącego z jednego z powyższych urządzeń należy dołączyć wtyczkę sieciową. Jeżeli użyta będzie wtyczka 13-amperowa (BS 1363), to bezpiecznik należy zmienić na 5-amperowy. Bezpiecznik 13-amperowy wewnątrz wtyczki nie powinien być używany. Jeżeli użyta będzie inna wtyczka, trzeba ją zaopatrzyć w bezpiecznik 5-amperowy, bądź umieścić go w tablicy rozdzielczej obok gniazdek sieciowych (1)

Ważne

Przewody w kablu sieciowym są oznaczone kolorami wg następującego kodu:

niebieski - przewód obojętny
brązowy - przewód aktywny

Ponieważ kod ten może różnić się od kodu stosowanego we wtyczce, należy postąpić w taki sposób:

Przewód oznaczony kolorem niebieskim musi być podłączony do końcówki z literą N lub czarnej.

Przewód oznaczony kolorem brązowym winien być połączony z końcówką z literą L lub czerwonej.

(1) W Wielkiej Brytanii stosuje się wtyczki, w które wbudowany jest bezpiecznik. Określają to normy Brytyjskie (np. BS 1363-British Standard 1363 i in.). Uwagi te nie mają zastosowania w Polsce - przyp. tłum.

Nie wolno usuwać żadnych śrubek ani próbować otworzyć obudowę urządzeń. Zawsze należy mieć na uwadze ostrzeżenie na plakietce umieszczone pod spodem CPC 6128 i MP2 oraz z tyłu GT 65 i CTM 664.

OSTRZEŻENIE: ELEMENTY POD NAPIĘCIEM. NIE USUWAĆ ŻADNYCH ŚRUB

Połączenie komputera z monitorem

(jeżeli używasz CPC 6128 z modulatorem/zasilaczem - przejdź do następnego paragrafu).

1. Upewnij się, że monitor nie jest włączony do sieci
2. Połącz przewód wychodzący z monitora i zakończony dużą wtyczką typu DIN i 6 wyprowadzeniami z gniazdem oznaczonym z tyłu komputera napisem MONITOR
3. Połącz przewód wychodzący z monitora i zakończony małą wtyczką (5V DC) z gniazdem z tyłu komputera oznaczonym 5V DC
4. Połącz przewód wychodzący z komputera i zakończony małą wtyczką (12 V DC) z gniazdem z przodu monitora.

Połączenie komputera z modulatorem/zasilaczem MP2

MP2 jest urządzeniem, które być może zechcesz nabyć, jeżeli używasz CPC 6128 z zielonym monitorem GT 65. MP2 pozwala na dołączenie Twojego komputera do domowego odbiornika telewizji kolorowej - a co za tym idzie - wykorzystanie pełnych możliwości kolorowej grafiki dostępnej w CPC 6128.

MP2 powinien być umieszczony po prawej stronie CPC 6128.

1. Upewnij się, że MP2 nie jest włączony do sieci.
2. Połącz przewód wychodzący z MP2 i zakończony dużą wtyczką typu DIN o sześciu końcówkach z gniazdem z tyłu komputera oznaczonym MONITOR
3. Połącz przewód wychodzący z MP2 i zakończony małą wtyczką (5V DC) z gniazdem z tyłu komputera oznaczonym 5V DC
4. Połącz przewód wychodzący z MP2 i zakończony wtyczką antenową z gniazdkiem oznaczonym AERIAL lub ANTENA w odbiorniku TV.

5. Połącz przewód wychodzący z komputera i zakończony małą wtyczką (12V DC) z gniazdkiem umiejscowionym z tyłu MP2.

Włączanie systemu - CPC 6128 i GT 65/CTM 664

(Jeżeli używasz CPC 6128 z zasilaczem/modulatorem MP2 - przejdź do następnej sekcji tego rozdziału).

Połączywszy cały system tak, jak to zostało opisane, włącz wtyczkę sieciową do gniazdka. Naciśnij teraz przycisk POWER po prawej stronie na dole monitora tak, aby przycisk był w położeniu "w c i ś n i ę t e". Kiedy przycisk jest w położeniu "w y c i ś n i ę t e" - zasilanie jest odłączone.

Włącz teraz komputer przy pomocy przełącznika na tylnej ściance, po prawej stronie. Przełącznik oznaczony jest POWER.

Czerwona lampka w środku górnej ścianki komputera powinna się rozjaśnić a na ekranie pojawi się napis

```
Amstrad 128K Microcomputer  (v3)
(C) 1985 Amstrad Consumer Electronics plc
         and Locomotive Software Ltd.
BASIC 1.1
Ready
```

Aby uniknąć zmęczenia oczu należy wyregulować jasność ekranu pokrętłem BRIGHTNESS. Jasność winna być odpowiednia dla wygodnej pracy a litery nie powinny być zamazane lub zbyt ostre.

Pokrętło BRIGHTNESS znajduje się z przodu monitora GT 65 i na prawej bocznej ściance CTM 664.

Jeżeli używasz monitora GT 65 może zajść potrzeba wyregulowania kontrastu pokrętłem CONTRAST i synchronizacji pionowej pokrętłem Vertical HOLD. Pokrętła te znajdują się na dole, na przedniej ściance.

Kontrast - pokrętłem CONTRAST - powinien być zmniejszony do możliwego minimum.

Monitor GT 65 winien być wyregulowany pokrętłem Vertical HOLD (oznaczonym V-HOLD) tak, aby obraz na ekranie był stabilny i nie występowało "pływanie" lub "drżenie" obrazu.

Włączenie systemu - CPC 6128 i MP2

Po połączeniu systemu w opisany uprzednio sposób, należy włączyć wtyczkę sieciową do gniazda zasilania.

Należy teraz włączyć komputer przełącznikiem POWER znajdującym się na tylnej ściance po prawej stronie.

Rozjaśni się lampka koloru czerwonego w środku górnej ścianki komputera i należy teraz dostroić odbiornik TV do częstotliwości sygnału generowanego przez komputer.

Jeżeli Twój odbiornik TV ma klawisze wyboru kanału, to należy wybrać wolny lub nieużywany kanał. Następnie, przy pomocy pokrętła płynnej regulacji należy dostroić telewizor do mniej więcej 36 kanału - tak, aby na ekranie telewizora ukazał się stabilny i wyraźny napis

Amstrad 128K Microcomputer (v3)
C 1985 Amstrad Consumer Electronics plc
and Locomotive Software Ltd

BASIC 1.1
Ready

Napis na ekranie będzie złotożółty na granatowym tle.

Jeżeli Twój telewizor ma bębnowy przełącznik kanałów to należy go obracać aż do uzyskania stabilnego obrazu. (Znów będzie to w okolicach kanału 36).

Inne połączenia

Jeżeli pragniemy dołączyć inne urządzenia zewnętrzne, czyli:

- Joystick'i
- Magnetofon
- Drukarkę
- Drugi napęd dysków
- Zewnętrzny wzmacniacz/głośniki
- Urządzenia rozszerzające

to szczegóły ich podłączenia znajdują się w części 2 rozdziału "Kurs podstawowy".

Na koniec, upewnij się, że zastosowałeś się do uwag podanych na początku tego podręcznika w części "WAŻNE".

Uwagi o instalacji 1,2,4,5,6,7,8

Uwagi o obsłudze 1

Część 2: Dołączanie urządzeń zewnętrznych

Ta część rozdziału opisuje, w jaki sposób należy dołączać do CPC 6128 różne urządzenia zewnętrzne i przystawki. Szczegóły dotyczące ich użycia będą podane w odpowiednich sekcjach tego podręcznika.

Joystick

Jeżeli używasz gier, które mają możliwość współpracy z joystickiem, być może zechcesz nabyć MASOFT joystick model JY2 - jedno z urządzeń dołączanych do CPC 6128.

Wtyczkę, którą zakończony jest kabel wychodzący z JY2 należy wetknąć do gniazda w komputerze oznaczonego JOYSTICK. CPC 6128 może współpracować z dwoma joystickami; drugi joystick należy połączyć z odpowiednim gniazdem umieszczonym w obudowie pierwszego.

Również joystick typu AMSOFT JY1 może być połączony z CPC 6128.

Dalsze informacje dotyczące joysticków znaleźć można w następnych rozdziałach.

Magnetofon

Zamiast dysku, do zapisywania i odtwarzania programów można użyć magnetofonu. Komendy, które powodują załadowanie programów na taśmę lub dysk oraz przepisujące programy z taśmy lub dysku do pamięci zostaną opisane dalej.

W celu dołączenia do CPC 6128 magnetofonu kasetowego niezbędny będzie przewód oznaczony AMSOFT CL1, lub jakikolwiek inny typowy przewód magnetofonowy.

Koniec przewodu z większą, 5-bolcową wtyczką należy połączyć z gniazdem komputera oznaczonym TAPE.

Następnie należy połączyć - końcówkę niebieskiego przewodu z gniazdkiem magnetofonu oznaczonym REMOTE lub REM; końcówkę czerwonego przewodu z gniazdkiem magnetofonu oznaczonym MIC, COMPUTER IN lub INPUT; końcówkę białego przewodu z gniazdkiem magnetofonu oznaczonym EAR, COMPUTER OUT lub OUTPUT.

Należy pamiętać, że prawidłowe przepisanie danych między komputerem i magnetofonem zależy w sposób istotny od właściwego ustawienia pokręteł LEVEL lub VOLUME magnetofonu. Jeżeli zapisy bądź odczyty programów nie udają się - należy eksperymentalnie dobrać optymalne ustawienie nastaw.

Drukarka

CPC 6128 może współpracować z dowolną drukarką o łączu równoległym typu CENTRONICS. Jeżeli używać chcemy drukarki AMSTRAD DMP1, należy po prostu użyć przewodu dostarczonego wraz z drukarką.

Jeżeli używana ma być inna drukarka o łączu CENTRONICS, to potrzebny będzie przewód łączący MASOFT PL1.

Koniec przewodu zaopatrzony w duże złącze krawędziowe należy połączyć z gniazdem oznaczonym w komputerze napisem PRNTER (z tyłu komputera).

Drugi koniec przewodu, zakończony złączem typu używanego przez drukarki z interfejsem CENTRONICS, należy połączyć z gniazdem drukarki. Jeżeli gniazdo drukarki wyposażone jest w zatrzaski zabezpieczające przed rozłączeniem się kabla - można je założyć na wtyczkę kabla, w wycięcia przeznaczone do tego celu.

Szczegóły działania drukarki znajdują się w dalszej części podręcznika.

Drugi napęd dysków (AMSTRAD FD1)

Urządzenie AMSTRAD FD1 może być dołączone do komputera jako drugi napęd dysków. Przewaga posiadania 2 napędów w systemie jest szczególnie oczywista dla programistów pracujących

z systemem CP/M; wiele programów jest tak zorganizowanych, że zbiory biblioteczne winny się znajdować na dysku włożonym do jednego napędu, a robocze zbiory danych zapisywane są na drugim dysku.

Kiedy FD1 jest już połączony z 6128 WPIERW włącz dodatkowy napęd FD1 - przy pomocy włącznika z tyłu urządzenia - a NASTĘPNIE komputer przy pomocy włącznika z tyłu komputera. W FD1 winny się zapalić oba wskaźniki - czerwony i zielony. Wtedy to cały 2-dyskowy system jest gotowy do pracy.

Szczegóły pracy z systemem 2-dyskowym znajdują się w dalszej części podręcznika.

Dodatkowe wzmacniacze i głośniki

System 6128 może być podłączony do dodatkowego wzmacniacza akustycznego stereo i kolumn głośnikowych. Pozwoli to na wykorzystanie pełnych możliwości dźwiękowych 3 kanałów komputera.

Przewód łączący wzmacniacz z komputerem powinien być zakończony 3,5 mm wtyczką typu "jack"-stereo. Wtyczkę tę należy połączyć z gniazdem oznaczonym na komputerze "STEREO".

Wyprowadzenia są jak następuje:

- zakończenie wtyczki	- kanał lewy
- środkowy pierścień	- kanał prawy
- nasada	- masa

6128 generuje sygnał o stałym poziomie, zatem należy wykorzystywać pokrętła wzmacniacza. Dotyczy to głośności, barwy dźwięku i balansu.

Słuchawki wysokoomowe mogą również współpracować z komputerem, jednakże natężenie dźwięku nie będzie regulowane przez pokrętło VOLUME w komputerze. Słuchawki niskoomowe, takie, jakie zwykle stosuje się w sprzęcie HI-FI nie mogą bezpośrednio współpracować z systemem.

Szczegóły dotyczące sterowania przepływem dźwięków przez poszczególne kanały opisane będą w dalszej części podręcznika.

Inne urządzenia

Inne urządzenia jak:interfejsy szeregowe, modemy, pióra świetlne, dodatkowe pamięci ROM itd. mogą być dołączone do 6128 poprzez gniazdo oznaczone EXPANSION, a umieszczone z tyłu komputera.

W ten sposób może być również dołączony syntezator mowy i wzmacniacz AMSOFT SSA2.

Szczegóły dołączania urządzeń poprzez gniazdoEXPANSION będą omówione w rozdziale "Trochę użytecznych informacji".

Na koniec upewnij się, czy zastosowałeś się do wskazań zawartych na początku podręcznika, w części "WAŻNE":

Uwagi o instalacji 6,7

Uwagi o obsłudze 4,8

Część 3: O dyskach

Komputer AMSTRAD CPC 6128 wykorzystuje 3 calowe dyski typu "compact". Zdecydowanie polecamy używanie wyłącznie dysków AMSOFT CF2, dających gwarancję pewnej pracy. Jednakże można również używać dysków wyprodukowanych przez inne wiodące firmy.

Wkładanie dysku

Każda ze stron dysku może być używana niezależnie. Dysk należy wkładać nalepką na zewnątrz; ta strona dysku, którą chcemy wykorzystać powinna być odwrócona ku górze.

Zabezpieczenie przed zapisem

Z lewej strony każdego dysku znajduje się strzałka wskazująca na mały, zasłaniany otwór. Określany on jest jako "otwór zabezpieczający przed zapisem", pozwalający na zabezpieczenie przed wymaganiem lub zniszczeniem danych.

Jeżeli otwór ten jest zasłonięty - dane mogą być zapisane na dysk przez komputer. Jeżeli jest on odkryty - dysk nie zezwala na dokonywanie na nim zapisu; pozwala to na uniknięcie przypadkowego zniszczenia cennych programów.

Różni wytwórcy dysków stosują różne mechanizmy zasłaniania otworu. Czynność odsłaniania otworu w dysku AMSOFT CF2 polega na:

- przesunięciu małej zasłonki umieszczonej w lewym rogu dysku.

Aby zasłonić otwór należy:

- przesunąć zasłonkę na jej pierwotne miejsce.

Niektóre inne dyski mają mały plastikowy bloczek umieszczony w szczelinie po lewej stronie dysku.

W celu odsłonięcia otworu należy ten bloczek przesunąć w stronę środka dysku, używając długopisu lub innego cienkiego przedmiotu.

Zwróćmy uwagę, że niezależnie od użytego sposobu otwierania i zamykania, otwarcie otworu zawsze zabezpiecza przed zapisem.

WAŻNE

Dopilnuj, aby otwory zabezpieczające w dyskach stanowiących pakiet systemowy CP/M były zawsze otwarte.

Kiedy dysk jest już włożony

Z przodu mechanizmu napędu dysków w komputerze widoczne są: czerwona lampka i przycisk do wyjmowania dysków.

Lampka

Lampka pokazuje, że na dysku odbywa się operacja zapisu bądź odczytu.

Jeżeli podłączony jest drugi napęd dyskowy, to lampka tego napędu będzie się stale świecić. Będzie ona przygasać w czasie, gdy komputer będzie używać głównego napędu.

Przycisk

Naciśnięcie tego przycisku pozwala na wyjęcie dysku z napędu.

Na koniec należy się upewnić, czy zastosowano się do następujących wskazówek zawartych na początku tego rozdziału, w części zatytułowanej "WAŻNE":

Uwagi o obsłudze 1,3,4,5,6.

Część 4: Zaczynamy ...

Zanim zaczniemy ładować programy do pamięci lub zapisywać je na dysk, zapoznajmy się z niektórymi klawiszami komputera. Ci spośród Państwa, którzy mają już te doświadczenia za sobą, mogą ominąć niniejszy rozdział.

Z już włączonym do sieci komputerem i ekranem, na którym wyświetlony jest nagłówek przystąpimy do poznawania funkcji różnych klawiszy.

Klawisze kursora ↑ ↓ ← → (po prawej stronie na dole klawiatury) zmieniają pozycje kursora (czyli małego kwadracíku) na ekranie.

Naciśnij każdy z tych klawiszy a zobaczysz, jak kursor przesuwa się po ekranie.

Klawisz [RETURN] wprowadza napisaną przez Ciebie informację do komputera. Po każdym naciśnięciu tego klawisza następuje przejście do nowej linii na ekranie. Każde polecenie wprowadzane do komputera powinno kończyć się naciśnięciem klawisza [RETURN].

Od tej chwili, poprzez symbol [RETURN] oznaczać będziemy "naciśnij klawisz RETURN po każdej komendzie lub linii programu.

W czasie normalnej pracy klawisz [ENTER] działa tak samo jak [RETURN] i może być używany zamiennie. Jednakże należy pamiętać, że podobnie jak klawisze części numerycznej, klawisz ten może być przedefiniowany i nabrać innego znaczenia. Powrócimy do tej sprawy później.

Klawisz [DEL] - służy do kasowania znaku na lewo od kursora na ekranie.

Napisz na ekranie a b c d a przekonasz się, że litera d znajduje się na lewo od kursora. Jeżeli dojdziesz do wniosku, że litera d jest zbędna, to naciśnij klawisz [DEL] a zniknie ona z ekranu. Jeżeli dalej będziesz naciskał [DEL], to zniknie cały napis.

Są dwa klawisze [SHIFT].
Jeżeli przyciśniesz którykolwiek z nich równocześnie z klawiszem litery/cyfry to na ekranie pojawi się duża litera lub symbol napisany na górze klawisza.

Naciśnij klawisz e, a następnie naciśnij klawisz [SHIFT] i trzymając go naciśnij klawisz e raz jeszcze. Na ekranie pojawi się

eE

Wprowadź teraz kilka spacji. Używając klawiszy z cyframi (najwyższy rząd klawiszy) napisz na ekranie 2 a następnie powtórnie naciśnij klawisz 2 jednocześnie trzymając klawisz [SHIFT].
Na ekranie zobaczysz

2"

Możesz przekonać się, co dzieje się na ekranie gdy naciskasz różne klawisze jednocześnie z SHIFT . Prosimy przeprowadzić różne próby naciskając klawisze raz z - a powtórnie bez klawisza [SHIFT].

Klawisz [CAPS LOCK] ma podobną funkcję do [SHIFT] z tym, że należy go nacisnąć tylko raz. Od tej chwili wszystkie litery pojawią się jako litery drukowane (duże, wersaliki) pomimo, że klawisze z cyframi nie zmienią swojego znaczenia.

Wciśnij [CAPS LOCK] i napisz

a b c d e f 1 2 3 4 5 6

a na ekranie zobaczysz

A B C D E F 1 2 3 4 5 6

Zauważysz, że klawisze z cyframi nie spowodowały wyprowadzenia na ekran symboli. Jeżeli pragniesz wprowadzić symbole w czasie, gdy działa klawisz [CAPS LOCK] - naciśnij klawisz [SHIFT]. Napisz, trzymając jednocześnie klawisz SHIFT :

a b c d e f 1 2 3 4 5 6

Na ekranie zobaczysz

A B C D E F ! " # $ %

Jeżeli chcesz powrócić do pisania małymi literami to po prostu powtórnie naciśnij klawisz [CAPS LOCK].

Jeżeli chcesz pisać dużymi literami i wprowadzać symbole bez konieczności ciągłego naciskania klawisza [SHIFT] to można to uzyskać poprzez wciśnięcie klawisza [CONTROL] i [CAPS LOCK] jednocześnie Wprowadza to tryb pracy "SHIFT LOCK". Wpisz w ekran

a b c d e f 1 2 3 4 5 6

Na ekranie zobaczysz

A B C D E F ! " # $ %

Prosimy zwrócić uwagę, że cyfry można wpisywać w dalszym ciągu używając klawiszy z części numerycznej klawiatury (fØ do f9).

Przytrzymanie klawisza [CONTROL] i jednokrotne naciśnięcie [CAPS LOCK] przywróci poprzedni tryb pracy klawiatury,(tj. albo tylko małe albo tylko duże litery). Jeżeli powróciłeś do trybu "tylko duże litery" - to po prostu naciśnij jeden raz klawisz [CAPS LOCK] a przejdziesz do pisania małymi literami.

Klawisz [CLR] służy do zamazywania znaku pod kursorem.

Wprowadź ABCDEFGH. Kursor umiejscowiony będzie na prawo od ostatniej wpisanej litery (H). Naciśnij klawisz ← cztery razy. Kursor przesunie się o cztery pozycje w lewo i będzie nałożony na literę E.

Zauważ, że litera E widoczna jest pod kursorem. Naciśnij teraz klawisz [CLR] jeden raz a zobaczysz, że litera E zniknie, a litery FGH przesuną się o jedną pozycję w lewo, z literą F ustawioną pod kursorem. Naciśnij teraz [CLR] i przytrzymaj klawisz. Zobaczysz, jak wymazywana jest litera F, a za nią G i F.

Klawisz [ESC] służy do przerwania (ang. ESCape) działania realizowanej przez komputer funkcji. Jednokrotne naciśnięcie [ESC] powoduje zawieszenie wykonywania programu. Może on być wznowiony po naciśnięciu innego klawisza.

Dwukrotne naciśnięcie klawisza [ESC] spowoduje permanentne przerwanie działania programu. Komputer przyjmować zacznie wtedy informacje w trybie pisania programów.

Kiedy już wpisywane litery osiągną prawy skraj ekranu, co ma miejsce po wprowadzeniu 40 znaków, to następny symbol pojawi się automatycznie przy lewym brzegu w następnej linijce. Oznacza to, że NIE należy naciskać [RETURN], co może wydać się konieczne tym z Państwa, którzy przyzwyczajeni są do maszyn do pisania.

Funkcję taką komputer wykonuje automatycznie. Na wprowadzony nadmiarowo [RETURN] odpowie komunikatem o błędzie - albo natychmiast - albo w czasie wykonywania programu. Najczęściej będzie to komunikat:

Syntax error

Syntax error /Błąd składni/

Jeżeli na ekranie pojawi się komunikat: Synatax error, to oznacza to, że komputer sygnalizuje, iż nie rozumie wprowadzonego polecenia.

Dla przykładu napisz

```
printt [RETURN]
```

Na ekranie zobaczysz napis

```
Syntax error
```

Stało się tak, bowiem komputer nie zrozumiał instrukcji:

```
printt
```

Jeżeli błędne polecenie wpisane zostanie w linijkę programu, tak jak np.

```
1Ø printt "abc" [RETURN]
```

to komunikat Syntax error pojawi się dopiero w czasie wykonywania programu.

Wpisz

```
run [RETURN]
```

/Polecenie to nakazuje wykonanie dopiero co wprowadzonego do pamięci programu/.

Na ekranie zobaczysz:

    Syntax error in 1Ø
    1Ø printt "abc"

Komputer w ten sposób informuje, że w linii 1Ø programu wystąpił błąd składni oraz wyświetla błędną linię z kursorem ustawionym na pierwszy znak tej linii.

Naciśnij klawisz → tyle razy, aby znalazł się on na literze t w słowie printt. Następnie naciśnij klawisz [CLR] i tym samym usuń zbędną literę. Teraz naciśnij klawisz [RETURN] sygnalizując koniec poprawiania.

Napisz:

    run [RETURN]

a zobaczysz, że komputer zaakceptował polecenie i wydrukował (ang. "print"/

    abc

Na koniec upewnij się, że zastosowałeś się do ostrzeżeń podanych na początku książki, w sekcji zatytułowanej "WAŻNE":

Uwagi o instalacji 4,5
Uwagi o obsłudze 1

Część 5: Ładowanie programów i gier

Witamy tych, którzy przeskoczyli tutaj z początku poprzedniej części!

Aby szybko przekonać się o prędkości, z jaką ładowane jest do pamięci oprogramowanie rezydujące na dysku włącz system i wsuń do napędu 4 stronę pakietu dysków systemowych CP/M (tj. stronę oznaczoną 4 do góry).

Napisz

run"rointime.dem" [RETURN]

Po kilku sekundach program będzie już załadowany do pamięci. Odpowiedz na postawione na ekranie pytanie o to, czy używasz zielonego monitora (Naciśnij Y jeżeli tak, N jeżeli nie) a zobaczysz na ekranie ciągłą demonstrację gry "Roland in time". Być może skłoni Cię ona nawet do zakupienia kopii tej gry!

Kiedy już skończysz oglądanie tego pokazu, to możesz "uciec" z programu przez przyciśnięcie klawisza [ESC] przy jednoczesnym

wciskaniu klawiszy [CONTROL] i [SHIFT]. Daje to efekt całkowitego wyzerowania komputera i może być używane, gdy pragniesz rozpocząć pracę ze stanu jałowego. (Nie ma potrzeby wyjmowania dysku z napędu resetując-zerując komputer w ten sposób).

Jeżeli program nie załaduje się, to w celu dowiedzenia się przyczyny błędu należy przestudiować informację o błędzie ukazującą się na ekranie:

Drive A: disc missing

Retry, Ignore or Cancel?

... oznacza, że albo niewłaściwie wprowadziłeś dysk do napędu, albo - o ile masz system z 2 napędami - włożyłeś go do napędu B

ROINTIME.DEM not found

... oznacza, że albo włożyłeś niewłaściwy dysk,(lub złą jego stronę), albo niezbyt precyzyjnie napisałeś nazwę programu, ROINTIME.DEM

Bad command

... oznacza, że prawdopodobnie źle napisałeś ROINTIME.DEM wprowadzając do nazwy zbędną spację lub znak przestankowy.

Type mismatch

... oznacza, że opuściłeś znak cudzysłowu "

Syntax error

... oznacza, że źle napisałeś słowo run

Drive A: read fail

Retry, Ignore or cancel

... oznacza, że komputerowi nie udało się odczytać danych z Twojego dysku. Sprawdź, czy włożyłeś właściwy dysk i naciśnij R ("Ponów próbę" - retry). Komunikat powyższy ukaże się, jeżeli kiedykolwiek zdarzyło Ci się popsuć dysk wyłączając komputer z dyskiem włożonym do napędu.

Kiedy już dowiesz się, jak wykonywać kopie dysków - zawsze zabezpieczaj cenne programy przez wykonanie ich kopii. Dotyczy to szczególnie wzorcowego pakietu dysków systemowych CP/M.

Ładowanie oprogramowania AMSOFT i program "WELCOME"

Wyostrzywszy sobie apetyt na milszą stronę używania komputera, załadujmy grę.

Wprowadź swój dysk z programem do napędu i napisz

run "disc" [RETURN]

Po kilku sekundach Twoja gra będzie załadowana i gotowa do zabawy.

Spróbuj napisać: run "disc", kiedy w napędzie jest 4 strona pakietu dysków systemowych CP/M, a zobaczysz i usłyszysz nieprzerwaną demonstrację programu "Welcome" (Witajcie).

Kiedy już skończysz oglądanie programu, wyzeruj komputer używając klawiszy [CONTROL] [SHIFT] i [ESC].

Powyższa instrukcja (run "disc") załaduje większość programów firmy AMSOFT; możesz jednak napotkać przypadek, że będziesz musiał napisać coś dodatkowo. We wszystkich jednak przypadkach instrukcja ładowania programu znajdować się będzie na obwolucie dysku. Instrukcji tej należy pilnie przestrzegać.

Na zakończenie upewnij się, że zastosowałeś się do ostrzeżeń podanych na początku tego podręcznika w rozdziale "WAŻNE"

Uwagi o instalacji 6

Uwagi o obsłudze 1,5,6

Część 6: Policzmy trochę ...

Jak dotąd, wiemy co musimy zrobić z komputerem i czego nam nie wolno robić; wiemy jak go połączyć i jak dołączyć urządzenia zewnętrzne. Wiemy, jak działają niektóre klawisze komputera i jak załadować program. Teraz przyjrzymy się niektórym instrukcjom.

Tak jak Ty czy ja - komputer zrozumieć może tylko polecenia w języku który zna; język ten nazywa się BASIC (skrót od Beginners' All-purpose Symbolic Instruction Code). Słowa w słowniku języka BASIC zwane są "słowami kluczowymi" i każde z nich nakazuje komputerowi wykonanie specyficznej funkcji. Każdy język musi trzymać się zasad gramatyki - i BASIC nie jest pod tym względem wyjątkiem. Tutaj w odniesieniu do gramatyki używa się określenia "składnia" i komputer będzie na tyle miły, że zawsze nie omieszka poinformować Cię, jeżeli zrobiłeś "Błąd składni" (Syntax error).

Wprowadzenie do słów kluczowych języka BASIC

W rozdziale zatytułowanym "Kompletna lista ..." znajdziesz opis wszystkich słów kluczowych języka AMSTRAD BASIC. W tej części wprowadzimy niektóre najczęściej używane słowa kluczowe.

CLS

Aby wyzerować (oczyścić) ekran napisz:

CLS [RETURN]

Zauważysz, że ekran staje się pusty i tylko w górnym lewym rogu zostaje napis Ready wraz z kursorem.

Zwróć uwagę, że do wprowadzania słów kluczowych możesz używać zarówno dużych jak i małych liter.

PRINT

Jest używane, gdy chcemy w programie wydrukować literę, słowo lub znak. Wprowadź następującą linijkę polecenia:

print "czesc" [RETURN]

Na ekranie zobaczysz

CZESC

Cudzysłów " " używany jest do zakomunikowania komputerowi, co należy wydrukować.

Czesc pojawi się na ekranie zaraz po naciśnięciu klawisza [RETURN]. Napisz

cls [RETURN]

aby oczyścić ekran

RUN

Powyższy przykład pokazywał jedną linijkę instrukcji. Jednakże, zaraz po naciśnięciu klawisza [RETURN] instrukcja została wykonana i natychmiast zapomniana. Możliwe jest wpisanie do komputera ciągu instrukcji po to, aby zostały wykonane w odpowiedniej kolejności. Osiąga się to poprzez napisanie "programu". Instrukcje języka BASIC używane w programach są takie same jak te, które poznaliśmy z tym, że przed każdą z nich wstawia się liczbę. Jeżeli w programie jest więcej niż jedna linia, to liczby te (numery instrukcji) informują komputer o kolejności ich wykonywania (ang. "run"). Po naciśnięciu [RETURN] linijka taka zapamiętywana jest w pamięci aż do

chwili uruchomienia programu. Napisz teraz:

10 print "czesc" [RETURN]

Zwróć uwagę, że po naciśnięciu RETURN "czesc" nie zostało napisane na ekranie. Aby uruchomić ten program należy użyć słowa run. Teraz napisz

run [RETURN]

a zobaczysz słowo czesc na ekranie.

Zapamiętaj, że zamiast ciągle pisać słowo print możesz użyć zamiast niego znaku zapytania; dla przykładu

10 ? "czesc" [RETURN]

LIST

Po wpisaniu programu do pamięci można sprawdzić co naprawdę zostało wpisane przez "wylistowanie" programu. Napisz:

list [RETURN]

Na ekranie zobaczysz:

10 PRINT "czesc"

tj. program zapamiętany w pamięci.

Zwróć uwagę, że słowo PRINT jest teraz napisane dużymi literami. Oznacza to, że BASIC zaakceptował słowo PRINT jako znane sobie słowo kluczowe.

Napisz: cls aby oczyścić ekran. Zwróć uwagę, że pomimo, iż ekran po naciśnięciu [RETURN] zostaje wyzerowany, to Twój program nie będzie wymazany z pamięci.

GOTO

Słowo GOTO mówi komputerowi aby przejść z jednej linii do innej w celu albo ominięcia pewnej ilości linii, albo stworzenia pętli. Napisz:

10 print "czesc" [RETURN]

20 goto 10 [RETURN]

Teraz napisz

run [RETURN]

... a zobaczysz słowo "czesc" wypisywane nieprzerwanie jedno pod drugim po lewej stronie ekranu. Dzieje się tak dlatego, że linia 20 nakazuje przejść do linii 10 i wykonać opisane tam działanie.

Aby wstrzymać działanie tego programu - naciśnij jeden raz klawisz [ESC]. Aby wznowić program - należy nacisnąć dowolny inny klawisz.

Aby przerwać działanie programu i móc wpisać inne polecenia naciśnij dwukrotnie klawisz [ESC].

Teraz napisz

cls [RETURN]

... aby wyczyścić ekran.

Jeżeli chcesz zobaczyć wyraz czesc wypisywany ciągle w każdej linii tak, że pokryje cały ekran to napisz powyższy program dodając jednak po cudzysłowiu średnik (;).

Napisz

```
10 print "czesc"; [RETURN]
20 goto 10 [RETURN]
run [RETURN]
```

Średnik informuje komputer, że następna grupa znaków ma być pisana bezpośrednio za poprzednią (o ile tylko ta nowa grupa nie jest zbyt długa aby zmieścić się w tej samej linii).

Przerwij działanie programu naciskając dwa razy [ESC]. Napisz powtórnie linię 10, tym razem jednak użyj przecinka (,) zamiast średnika (;).

```
10 print "czesc", [RETURN]
run [RETURN]
```

Zobaczysz, że przecinek (,) poinstruował komputer żeby następną grupę znaków napisać o 13 kolumn dalej od pierwszej grupy. Jest to pomocne przy wyprowadzaniu informacji w osobnych kolumnach. Jednakże, jeżeli grupa znaków przekracza 12, to następna grupa przesunięta będzie o następne 13 pozycji po to, aby zawsze utrzymać spację między kolumnami.

Liczba 13 (kolumn) może być zmieniona poleceniem ZONE opisanym w dalszej części tego podręcznika.

I znów, aby przerwać działanie programu naciśnij dwa razy [ESC]. Aby całkowicie oczyścić pamięć komputera przytrzymaj klawisze [CONTROL] i [SHIFT] i naciśnij [ESC]. Komputer zostanie wyzerowany.

INPUT

Ta komenda używana jest, aby poinformować komputer, że winien poczekać na wprowadzenie informacji; na przykład odpowiedzi na pytanie.

Wprowadź:

```
10 input "ile masz lat"; wiek [RETURN]
20 print "nie wyglądasz na";wiek;"lat" [RETURN]
run [RETURN]
```

Na ekranie zobaczysz

ile masz lat?

Teraz wprowadź swój wiek i naciśnij [RETURN]. Jeżeli twój wiek wynosił 18, na ekranie zobaczysz

nie wyglądasz na 18 lat

Ten przykład pokazał użycie instrukcji input i zmiennej liczbowej. Słowo wiek zostało wprowadzone do pamięci na końcu linii 10 tak, że odtąd komputer łączy słowo wiek z liczbą wprowadzoną przez ciebie i napisze tę liczbę w miejscu słowa wiek w linii 20. Jakkolwiek użyliśmy nazwy wiek do nazwania zmiennej, to mogła to być dowolna litera, na przykład b.

Wyzeruj komputer aby wyczyścić pamięć.
(klawisze [CONTROL][SHIFT] i [ESC]). Jeżeli chciałeś wprowadzić jakieś litery (litery lub litery i cyfry), to na końcu nazwy zmiennej należało umieścić znak dolara ($). Ten typ zmiennych określany jest jako "zmienne łańcuchowe".

Wpisz następujący program. (Zwróć uwagę, że w linii 20 należy umieścić spację po literze c w czesc i przed literą m w mam).

```
10 input "jak masz na imie"; imie$ [RETURN]
20 print "czesc "; imie$; " mam na imie Ronald " [RETURN]
run [RETURN]
```

Na ekranie zobaczysz

jak masz na imie?

Napisz je, a następnie naciśnij [RETURN].
Jeżeli imię, które wprowadziłeś brzmiało Fred to na ekranie zobaczysz

czesc Fred mam na imie Ronald

Jakkolwiek na nazwę zmiennej łańcuchowej wybraliśmy imie$, to równie dobrze mogła to być litera, dla przykładu a$. Połączymy teraz dwa powyższe przykłady w jeden program.

Wyzeruj komputer naciskając [CONTROL] [SHIFT] i [ESC].
Wpisz co następuje:

```
 5 cls [RETURN]
10 input "jak masz na imie"; a$ [RETURN]
20 input "ile masz lat";b [RETURN]
30 print "musze powiedziec";a$;"że nie wygladasz na";b;"lat"
        [RETURN]
run [RETURN]
```

W tym programie użyliśmy 2 zmiennych; a$ dla imienia i b dla wieku. Na ekranie zobaczysz:

jak masz na imie?

Wprowadź teraz swoje imię (np. Fred) i naciśnij [RETURN]

Zostaniesz teraz zapytany:

ile masz lat?

Napisz swój wiek (np. 18) i naciśnij [RETURN]. Jeżeli masz na imię Fred i 18 lat to na ekranie zobaczysz:

musze powiedzieć Fred że nie wygladasz na 18 lat

Poprawianie programu

Jeżeli którakolwiek z linii programu została wpisana z błędami - co zaowocowało pojawieniem się napisu Syntax error lub innego komunikatu o błędzie - to istnieje możliwość poprawienia tej linii zamiast wpisywania jej od początku.

Aby to pokazać, wpiszmy poprzedni program z błędami:

```
 5  clss [RETURN]
10  input "jak masz na imie";a$ [RETURN]
20  input "ile masz lat";b [RETURN]
30  print "musze powiedzieć";a$;"że nie wygladasz na";b;
        "lat" [RETURN]
```

W programie są 3 błędy:

W linii 5 napisaliśmy clss zamiast cls

W linii 10 napisaliśmy mas zamiast masz

W linii 30 zapomnieliśmy o spacji między powiedziec a"

Są trzy metody poprawiania programu.

Pierwszy - to po prostu przepisać całą linijkę. Kiedy linia jest napisana od nowa i wprowadzona, to zastępuje ona w pamięci poprzednią linię o tym samym numerze.

Drugi - to poprawianie przy pomocy kursora edycyjnego.
Trzeci - to poprawianie kursorem kopiującym.

Poprawianie kursorem edycyjnym

Aby poprawić błąd w linii 5 napisz:

edit 5 [RETURN]

Linia 5 pojawi się pod linią 30 z kursorem nałożonym na literę c w clss.

Aby usunąć zbędną literę s naciskaj klawisz → aż kursor znajdzie się nad ostatnią literą s i naciśnij klawisz [CLR]. Zobaczysz, że s zniknie.

Naciśnij teraz [RETURN] a linia 5 zostanie poprawiona w pamięci. Napisz

list [RETURN]

aby sprawdzić, że linia 5 jest już poprawiona.

Komenda AUTO, opisana w dalszej części podręcznika, może być użyta do poprawiania pewnej ilości kolejnych linii w sposób podobny do metody opisanej w Poprawianiu kursorem kopiującym.

Poprawianie kursorem kopiującym

Kursor kopiujący jest jeszcze jednym kursorem (obok tego, który jest już na ekranie), który pojawia się, kiedy przyciśniesz dowolny klawisz kursora jednocześnie z klawiszem SHIFT . Oddziela się on wtedy od głównego kursora i może być niezależnie przesuwany po ekranie.

Aby poprawić błędy w liniach 10 i 30 naciśnij klawisz [SHIFT] a następnie klawisz ↑ aż do momentu, kiedy kursor kopiujący znajdzie się na samym początku linii 10. Zauważysz, że normalny kursor nie przesunął się, tak więc na ekranie znajdują się dwa kursory. Naciśnij teraz klawisz [COPY] tyle razy, żeby kursor kopiujący znalazł się nad spacją pomiędzy s i na. Zauważysz, że linia 10 jest przepisywana w ostatniej linii i zwykły kursor przesuwa się równolegle z kursorem kopiującym. Teraz naciśnij literę z. Pojawi się ona tylko w dolnej linii.

Główny kursor przesunął się a kursor kopiujący został na swoim miejscu. Naciśnij teraz [COPY] tak, aby cała linia 10

była przepisana. Naciśnij [RETURN] a poprawiona linia zostaje zapamiętana w pamięci. Kursor kopiujący znika, a główny kursor ustawia się pod nową linią 1Ø. Aby poprawić drugą pomyłkę przytrzymaj [SHIFT] i naciskaj klawisz ↑ tak długo, aż kursor kopiujący pojawi się na samym początku linii 3Ø.

Naciskaj [COPY] aż kursor znajdzie się nad cudzysłowem obok słowa p o w i e d z i e ć. Teraz naciśnij spację. Zostanie ona wstawiona do dolnej linii. Przytrzymaj teraz klawisz [COPY] do chwili, gdy cała 3Ø linia jest przepisana i naciśnij [RETURN].

Możesz teraz wylistować program w celu sprawdzenia, że program w pamięci został poprawiony. Napisz

list [RETURN]

UWAGA: Aby szybko przesunąć kursor (podczas poprawiania) do prawego bądź lewego skraju linii naciśnij [CONTROL] i raz klawisze kursora ← lub →

Teraz wyresetuj komputer (wyzeruj go) przez naciśnięcie [CONTROL] [SHIFT] i [ESC].

IF

Komendy IF oraz THEN nakazują komputerowi sprawdzenie określonego warunku a następnie podjęcie działania w zależności od wyniku testu. Dla przykładu w instrukcji

if 1+1 = 2 then print "correct" [RETURN]

... komputer sprawdzi warunek i podejmie akcję w zależności od rezultatu.

Słowo kluczowe ELSE może być użyte do poinformowania komendy IF THEN o tym, jakie działanie należy podjąć na wypadek nie spełnienia warunku; dla przykładu

if 1+1=Ø then print"dobrze" else print "zle" [RETURN]

Rozszerzymy teraz poprzedni program o polecenia if i then.

Wprowadź następujący program. Zauważ, że dodaliśmy dwa symbole < znaczy "mniej niż" i znajduje się obok litery M . > znaczy "więcej niż" i znajduje się obok klawisza <.

```
 5 cls [RETURN]
1Ø input "jak masz na imie";a$ [RETURN]
2Ø input "ile masz lat";wiek [RETURN]
30 if wiek<13 then 6Ø [RETURN]
```

```
40 if wiek < 20  then  70 [RETURN]
50 if wiek > 19  then  80 [RETURN]
60 print " Cóż "; a$ "; " nie jesteś jeszcze nastola-
tkiem w wieku "; wiek " lat " : end [RETURN]
70 print " Cóż "; a$; " jesteś nastolatkiem w wieku ";
wiek; " lat " : end [RETURN]
80 print " Cóż "; a$; " nie jesteś już nastolatkiem
w wieku ", wiek; " lat. " : end [RETURN]
```

Aby sprawdzić ten program napisz

list [RETURN]

Teraz napisz

run [RETURN]

Następnie odpowiedz na pytanie komputera i zobacz co się stanie.

Widzisz teraz, jaki efekt w programie ma użycie instrukcji IF i THEN. Słowo kluczowe END użyte jest w programie dosłownie w celu zakończenia działania programu. Jeżeli w linii 60 nie byłoby słowa END to program działał by dalej wykonując linie 70 i 80.

Podobnie, jeżeli w linii 70 nie byłoby słowa END to program działał by dalej wykonując linię 80. Dwukropek : przed słowem END rozdziela dwie lub więcej instrukcji w jednej linii programu. Dodaliśmy również linię 5 aby wyczyścić ekran na początku programu. Zawsze będziemy to robić od tej chwili aby programy wyglądały ładniej.

Wyzeruj teraz komputer naciskając [CONTROL] [SHIFT] i [ESC].

FOR i NEXT

Komendy FOR i NEXT używane są, jeżeli wykonujemy pewną operację pewną ilość razy. Instrukcje określające operacje powinny być zamknięte pętlą FOR i NEXT.

Napisz

```
5  cls [RETURN]
10 for a=1 to 10 [RETURN]
20 print "numer kolejny"; a [RETURN]
30 next a [RETURN]
run [RETURN]
```

Zobaczysz, że operacja z linii 2Ø została wykonana 10 razy tak, jak zostało to ustalone instrukcją FOR w linii 1Ø. Zwróć uwagę, że wartość zmiennej a również została zwiększona 10 razy o 1.

Słowo kluczowe STEP (ang. "krok") może być użyte w komendzie FOR NEX do określenia, jaki ma być krok w każdej operacji. Dla przykładu zmień linie 1Ø do postaci

```
1Ø for a=1 to 5Ø step 5 [RETURN]
run [RETURN]
```

Kroki ujemne mogą być również używane. Dla przykładu

```
1Ø for a=1ØØ to Ø step - 1Ø [RETURN]
run [RETURN]
```

REM

Rem jest skrótem od REMark ("Uwaga"). Instruuje to komputer, że należy zignorować resztę linii. Można użyć REM do przekazania informacji, dla przykładu o tytule programu, użyciu zmiennych itd. Dla przykładu:

```
1Ø REM Schwyć najezdzców [RETURN]
2Ø L=5:REM ilość zyc [RETURN]
```

Pojedynczy znak cudzysłowia (apostrof) może być użyty zamiast REM. (Znak ten wprowadza się trzymając [SHIFT] i klawisz 7). Dla przykładu:

```
1Ø'Schwyć najezdzców [RETURN]
2Ø L=5: ' ilosc zyc [RETURN]
```

GOSUB

Jeżeli w programie występuje zestaw instrukcji, które powtarzane są wiele razy (w różnych miejscach programu) to nie ma potrzeby w każdym miejscu w programie powtarzać całego zestawu; przeciwnie - cały zestaw może być ujęty w postaci "podprogramu" który może być "wywołany" instrukcją GOSUB z numerem linii. Koniec podprogramu oznaczany jest instrukcją RETURN. Po napotkaniu tej instrukcji komputer wróci do linii następującej po linii z której wywołany był podprogram.

/Dwa następne programy nie robią nic poza wypisaniem tekstu na ekranie i nie trzeba trudzić się wpisywaniem ich do pamięci. Zostały one załączone wyłącznie w celu pokazania jak podprogramy wykonują powtarzalne zadania/.

Dla przykładu w następującym programie:

```
10 MODE 2 [RETURN]
20 PRINT "This old man he played one" [RETURN]
30 PRINT "He played knick-knack on my drum" [RETURN]
40 PRINT "With a knick-knack paddy wack" [RETURN]
50 PRINT "Give a dog a bone" [RETURN]
60 PRINT "This old man came rolling home" [RETURN]
70 PRINT [RETURN]
80 PRINT "This old man he played two" [RETURN]
90 PRINT "He played knick-knack on my shoe" [RETURN]
100 PRINT "With a knick-knack paddy wack" [RETURN]
110 PRINT "Give a dog a bone" [RETURN]
120 PRINT "This old man came rolling home" [RETURN]
130 PRINT [RETURN]
140 PRINT "This old man he played three" [RETURN]
150 PRINT "He played knick-knack on my knee" [RETURN]
180 PRINT "With a knick-knack paddy wack" [RETURN]
190 PRINT "Give a dog a bone" [RETURN]
200 PRINT "This old man came rolling home" [RETURN]
210 PRINT [RETURN]
run [RETURN]
```

widać wiele linii, które musiały być powtórzone w różnych miejscach programu, jak na przykład refren w linii 180. Zamieńmy teraz refren na podprogram i dodajmy na końcu instrukcję RETURN. Teraz będziemy wywoływać podprogram używając instrukcji GOSUB 180, kiedy tylko będziemy go potrzebować. Program wygląda teraz tak:

```
10 MODE 2 [RETURN]
20 PRINT "This old man he played one" [RETURN]
30 PRINT "He played knick-knack on my drum" [RETURN]
40 GOSUB 180 [RETURN]
80 PRINT "This old man he played two" [RETURN]
90 PRINT "He played knick-knack on my shoe" [RETURN]
100 GOSUB 180 [RETURN]
140 PRINT "This old man he played three" [RETURN]
150 PRINT "He played knick-knack on my knee" [RETURN]
160 GOSUB 180 [RETURN]
170 END [RETURN]
180 PRINT "With a knick-knack paddy wack" [RETURN]
190 PRINT "Give a dog a bone" [RETURN]
200 PRINT "This old man came rolling home" [RETURN]
210 PRINT [RETURN]
220 RETURN [RETURN]
run [RETURN]
```

Czy zauważyliście, ile zaoszczędziliśmy sobie zbędnego pisania? Dobrze zaprojektowane podprogramy są podstawową częścią programowania. Prowadzą one do strukturalizacji programów i wyrabiają u programisty dobre nawyki.

Należy mieć na uwadze, że nie zawsze trzeba "wskakiwać" do tego samego miejsca podprogramu; to jest na sam początek. Podprogram napisany w liniach 5ØØ-8ØØ może być wywoływany przez: GOSUB 5ØØ, GOSUB 640 lub GOSUB 790.

Zwróć uwagę, że w powyższym programie instrukcja END jest użyta w linii 170. W przeciwnym razie program w sposób naturalny kontynuuje działanie po linii 160 i wykonuje linię 180, co nie może mieć miejsca za wyjątkiem wywołania przez GOSUB.

Prosta arytmetyka

Całkiem łatwo można użyć komputera jako kalkulatora. Aby to zrozumieć, prosimy prześledzić następujące przykłady. Będziemy używali znaku ? jako synonimu instrukcji PRINT. Odpowiedź ukaże się na ekranie zaraz po naciśnięciu klawisza [RETURN].

Dodawanie

Wprowadź:

```
? 3+3 [RETURN]
6
```

/zwróćmy uwagę, że nie należy pisać znaku równości =/

Wprowadź:

```
? 8+4 [RETURN]
  12
```

Odejmowanie

Wprowadź:

```
? 4-3 [RETURN]
  1
```

Wprowadź:

```
? 8-4 [RETURN]
  4
```

Mnożenie

Wprowadź:

```
? 3*3 [RETURN]
  9
```

Wprowadź:

```
? 8*4 [RETURN]
  32
```

Dzielenie

Wprowadź:

```
? 3/3 [RETURN]
  1
```

Wprowadź:

```
? 8/4 [RETURN]
  2
```

Dzielenie całkowite

(użyj znaku \ jako symbolu dzielenia z odrzuceniem reszty)

Wprowadź:

```
? 10\6 [RETURN]
  1
```

Wprowadź

```
? 20\3 [RETURN]
  6
```

Reszta z dzielenia
(napisz MOD jako określenie tej operacji)
Wprowadź:

? 10 MOD 4 [RETURN]
2

Wprowadź:

? 9 MOD 3 [RETURN]
Ø

Pierwiastek kwadratowy
Aby znaleźć pierwiastek kwadratowy danej liczby napisz sqr( ).
Liczba podpierwiastkowa winna być wewnątrz nawiasów.
Wprowadź:

? sqr(16) [RETURN] (to oznacza $\sqrt{16}$)
4

Wprowadź:

? sqr(1ØØ) [RETURN]
1Ø

Podnoszenie do potęgi

Podnoszenie do potęgi to np. 3 do kwadratu (3↑2), 3 do trzeciej potęgi (3↑3) itp.
Wprowadź:

? 3↑3 RETURN (co oznacza 3^3)
27

Wprowadź:

? 8↑4 RETURN (co oznacza 8^4)
4096

Pierwiastek sześcienny
Pierwiastek sześcienny można bardzo łatwo wyliczyć metodą podobną do poprzedniej.
Aby znaleźć pierwiastek sześcienny z 27 ($\sqrt[3]{27}$) wprowadź:

? 27↑(1/3) RETURN
3

Aby uzyskać pierwiastek 3-go stopnia z 125 wprowadź:

? 125↑(1/3) RETURN
5

Obliczenia mieszane
(+,-,*,/)

Obliczenia mieszane są rozumiane przez komputer, lecz wykonywane są według określonych priorytetów. Najwyższy priorytet dany jest operacjom mnożenia i dzielenia, następnie dodawania i odejmowania. Tyczy się to obliczeń zawierających tyłko te cztery działania.

Jeżeli obliczamy

3+7-2*7/4

to można by przypuszczać, że obliczenia przebiegną tak:

3+7-2*7/4
= 8*7/4
= 56/4
= 14

W rzeczywistości jednak jest to obliczane jak następuje:

3+7-2*7/4
= 3+7-14/4
= 3+7-3.5
= 10-3.5
= 6.5

Przekonaj się o tym wykonując tak jak jest napisane:

```
? 3+7-2*7/4 [RETURN]
  6.5
```

Sposób obliczeń można zmienić wprowadzając nawiasy. Komputer będzie wykonywał obliczenia zawarte w nawiasach przed mnożeniem itp. poza nawiasami. Sprawdź to, wprowadzając obliczenie zawierające nawiasy:

```
?(3+7-2)*7/4 [RETURN]
 14
```

Priorytety wszystkich operatorów arytmetycznych są następujące (w kolejności malejącej):

	Potęgowanie
MOD	Reszta z dzielenia
-	Minus jednoargumentowy (wskazujący na liczbę ujemną)
* i /	Mnożenie i dzielenie
\	Dzielenie całkowite
+ i -	Dodawanie i odejmowanie

Dalsze wykładniki

Jeżeli chcesz w swoich obliczeniach używać liczb bardzo dużych lub bardzo małych, to wygodnie jest stosować zapis wykładniczy. Litera E używana jest do oznaczenia wykładnika liczby przy podstawie 10. Można używać E lub e.

Dla przykładu 300 to to samo co $3x10^2$. W notacji wykładniczej zapiszemy to jako 3E2. Podobnie, 0.03 to znaczy $3x10^{-2}$. W notacji wykładniczej odpowiada to 3E-2. Prześledź następujące przykłady.

Możesz wpisać:

```
? 3Ø*1Ø [RETURN]
  3ØØ
```

lub też

```
? 3E1*1E1 [RETURN]
  3ØØ

? 3ØØØ*1ØØØ [RETURN] lub ? 3E3*1E3 [RETURN]
  3ØØØØØØ

? 3ØØØ*Ø.ØØ1 [RETURN] lub ? 3E3*1E-3 [RETURN]
  3
```

Część 7: Zapisz program na dysk

Teraz, kiedy już przećwiczyłeś palce wpisując kilka instrukcji, chciałbyś pewnie wiedzieć jak przepisać program z komputera na dysk i jak załadować go z powrotem do pamięci.

Nawet jeżeli jesteś zaznajomiony z zapisywaniem i ładowaniem programów na/z kasety, to warto podać kilka informacji, które trzeba poznać mając do czynienia ze zbiorami dyskowymi.

Dwie główne różnice są następujące. Po pierwsze - na dopiero co wyjętą z opakowania dyskietkę nie można zapisać jeszcze żadnej informacji, co ma miejsce w wypadku kasety. Nowy dysk musi być wpierw "sformatowany", a czynność ta zostanie za chwilę przybliżona.

Drugi punkt wart wskazania w tym miejscu to waga, jaką należy przywiązywać do właściwego nadawania nazw zbiorom dyskowym. Nazwy zbiorów na kasetach zwykle muszą spełniać bardzo

luźne reguły, różniąc się bardzo swoją długością; czasami mogą być w ogóle pominięte.

Z dyskami jest zupełnie inaczej.

Nazwy zbiorów dyskowych muszą spełniać ściśle reguły narzucone przez CP/M; zostaną one podane dalej w tej części podręcznika.

Formatowanie dysków

Przed zapisaniem na dysku jakiejkolwiek informacji, musi on zostać sformatowany. Formatowanie może być porównane do budowania na dysku półek i przegródek przed wykorzystaniem ich do pamiętania informacji, innymi słowy do wyznaczenia zorganizowanej sieci wzdłuż której dane mogą być zapisywane lub odczytywane.

Formatowanie dzieli dysk na 360 rozłącznych obszarów

Na dysku jest 40 ścieżek ("tracks") - od najbardziej zewnętrznej (ścieżka Ø) do najbardziej wewnętrznej (ścieżka 39); obwód dysku podzielony jest na 9 sektorów ("sectors").

Każda ścieżka może w jednym sektorze zapamiętać 512 bajtów danych, stąd całkowita pojemność jednej strony dysku wynosi 180 KB (kilobajtów).

Pierwsze kroki przy używaniu wzorcowego pakietu dysków systemowych CP/M

Aby przygotować czystą dyskietkę do odczytywania i zapisywania Twoich programów, należy sformatować ją. Robi się to używając strony 1 pakietu dysków systemowych CP/M (dostarczonych z Twoim komputerem).

Włącz system, a następnie włóż stronę 1 pakietu do napędu w komputerze.

Jeżeli pracujesz z 2 napędami, zawsze wkładaj Stronę 1 do głównego napędu - tego wbudowanego do komputera. (Napęd A.)

Napisz:

|cpm RETURN

(Symbol | uzyskasz naciskając klawisz ∂ jednocześnie z [SHIFT].

Po kilku sekundach na górze ekranu zobaczysz następujący napis:

CP/M Plus Amstrad Computer Electronics plc

Jest to komunikat świadczący, że komputer znajduje się teraz pod kontrolą systemu operacyjnego CP/M. Zobaczysz również znaki A> wraz z kursorem. Są to tzw. "znaki gotowości" - podobne do Ready podczas normalnej pracy komputera w języku BASIC - i oznaczają, że komputer oczekuje na Twoje instrukcje.

Od chwili włączenia systemu CP/M nie możesz już wprowadzać komend języka BASIC, ponieważ nie będą one już rozumiane.

Jeżeli, dla przykładu, napiszesz komendę języka BASIC:

cls [RETURN]

to komputer przepisze Twoją komendę razem ze znakiem zapytania

CLS?

wskazując, że nie zrozumiał komendy.

Aby rzucić okiem na kilka z komend CP/M napisz

dir [RETURN]

Na ekranie zobaczysz skorowidz (DIRectory) komend systemowych i użytkowych. Jedną z nich jest komenda DISCKIT3.

Napisz:

disckit3 RETURN

Po chwili zobaczysz na górze ekranu nagłówek programu DISC KIT oraz poniżej

One drive found

Oznacza to, że działa program użytkowy DISC KIT oraz że komputer rozpoznał, że działa z jednym napędem dysków (tym wbudowanym do komputera)

Jeżeli podłączyłeś dodatkowy napęd do komputera, to napis będzie brzmiał:

Two drives found

Na dole ekranu zobaczysz co następuje

Copy	7
Format	4
Verify	1
Exit from program	Ø

Jest to tzw. "główne menu" programu. Numery w kwadratach odpowiadają klawiszom funkcyjnym po prawej stronie klawiatury (oznaczonymi fØ, f1, f4 i f7), a naciśnięcie jednego z nich odpowiada wybraniu jednej z opcji menu.

Prosimy zwrócić uwagę, że naciśnięcie w tej chwili klawisza f0 spowoduje opuszczenie programu DISC KIT i powrót do trybu wprowadzania bezpośredniego w systemie CP/M (znaki A>)

Chcemy teraz sformatować dysk, naciśnij zatem klawisz 4(f4)

OSTRZEŻENIE

FORMATOWANIE UPRZEDNIO ZAPISANEGO DYSKU SPOWODUJE WYMAZANIE JEGO ZAWARTOŚCI

Zobaczysz teraz na ekranie nowe menu, pozwalające na wybór różnych formatów, a dokładnie

System format	9
Data format	6
Vendor format	3
Exit menu	

Jak poprzednio możemy teraz nacisnąć jeden z klawiszy funkcyjnych (f3, f6, f9) i dokonać tym samym wyboru formatu. Wszystkie typy formatów zostaną objaśnione w dalszej części podręcznika, ale teraz wybierz "Data format" naciskając klawisz o numerze 6.

Zwróć uwagę, że naciśnięcie klawisza . (pod f9, f6 i f3) powoduje przejście z trybu formatowania do poziomu głównego menu programu DISC KIT.

Nacisnąwszy klawisz funkcyjny numer 6 (i zakładając że nie dołączyłeś do komputera dodatkowego napędu) zobaczysz informację:

Y Format as data
Any other key to exit menu

W tej chwili należy wyjąć dysk systemowy CP/M i włożyć dysk, który pragniesz sformatować. Strona dysku mająca zostać sformatowana powinna być zwrócona do góry napędu.

Teraz naciśnij klawisz Y (Y oznacza Yes-tak, kontynuuj działanie i sformatuj dysk).

Dysk zostanie sformatowany (po kolei ścieżki 0-39) a aktualny numer ścieżki wyświetlony będzie w lewym górnym rogu ekranu.

Nie można sformatować dysku, którego otwór zabezpieczający przed zapisem jest otwarty. Każda próba sformatowania takiego dysku łączyć się będzie z komunikatem:

Disc write-protected
Insert disc to format
R-etry or C-ancel

... po ukazaniu się którego należy nacisnąć C (jako Cancel - odwołaj komendę), wyjąć dysk, włożyć właściwy dysk do formatowania. Jego otwór zabezpieczający powinien być zamknięty.

Dopilnuj, aby otwór zabezpieczający w dyskach zawierających cenne programy zawsze był otwarty; NIGDY nie zamykaj tego otworu w dyskach, które składają się na wzorcowy pakiet dysków systemowych CP/M.

Kiedy formatowanie jest już zakończone, zostaniesz poproszony o wyjęcie dopiero co sformatowanego dysku i naciśnięcie dowolnego klawisza - co będzie sygnałem do kontynuowania programu.

Czynność formatowania można powtórzyć wsuwając nowy dysk i naciskając Y. Można tak formatować po kolei tyle dysków, ile chcemy mieć sformatowanych w danym formacie.

Kiedy już zakończysz formatowanie - naciśnij dowolny klawisz inny niż Y, a przejdziesz ponownie do poziomu głównego programu DISC KIT.

Działanie innych opcji programu, tj. C o p y i V e r i f y będą omówione w dalszej części podręcznika. Teraz, kiedy nauczyłeś się już formatować przy pomocy CP/M, wyzeruj komputer używając klawiszy [CONTROL] [SHIFT] i [ESC].

Zawsze trzymaj wzorcowe dyski CP/M w bezpiecznym miejscu; są one dosłownie kluczem do Twojego systemu. W dalszej części podręcznika dowiesz się, jak wykonać "robocze kopie" tych dysków po to, aby dyski wzorcowe móc trzymać w zamkniętym, pewnym schowku.

Formatowanie w systemie z 2 napędami

Idź za wskazówkami umieszczonymi powyżej, wybierz opcję Format (menu główne) naciskając klawisz f4 a następnie klawisz f6 wybierając Data format.

W tym miejscu zobaczysz trzecie menu, pozwalające wybrać napęd w którym ma się dokonać formatowanie.

Format A:	8
Format B:	6
Exit menu	2

Wybranie opcji Format B: (klawisz f5) pozwoli na pozostawienie Strony 1 dysku systemowego w napędzie A i umieszczenie dysku przeznaczonego do formatowania w napędzie B.

Po wybraniu "Format B:" możesz nacisnąć Y - co spowoduje rozpoczęcie formatowania - lub dowolny inny klawisz - aby przejść do głównego menu programu DISC KIT.

Jeżeli wybrałeś opcję "Format A:" (klawisz f8), to MUSISZ pamiętać, aby wyjąć dysk systemowy z napędu A: i włożyć dysk przeznaczony do formatowania.

Nie zapomnij:

NIGDY NIE RYZYKUJ ZAPISANIA DYSKÓW SYSTEMOWYCH CP/M.

Teraz, kiedy sformatowaliśmy jedną bądź dwie dyskietki, możemy zapisywać i odczytywać programy w języku BASIC na/z dysku.

Zapisywanie programów z pamięci na dysk

Po zapisaniu programu do pamięci komputera można przepisać go na dysk pisząc:

Save "nazwa zbioru" RETURN

Zwróć uwagę, że nadanie programowi nazwy jest obligatoryjne.

Nazwa zbioru na dysku składa się z dwu części (pól). Pierwsza część jest konieczna i może zawierać do 8 znaków. Można używać litery i cyfry, ale nie wolno używać spacji i znaków przestankowych. Część ta zawiera zwykle nazwę programu.

Druga część nazwy zbioru jest opcjonalna. Możesz użyć do 3 znaków; i tym razem nie są dozwolone spacje lub znaki przestankowe. Oba pola rozdzielone są znakiem kropki (.).

Jeżeli nie określisz drugiego pola, to system sam przyporządkuje mu odpowiedni ciąg znaków, taki jak .BAS dla oznaczenia programu w języku BASIC lub .BIN dla oznaczenia zbioru zawierającego kod maszynowy.

Jako przykład operacji zapisu na dysk napisz krótki program, następnie wsuń do napędu czysty sformatowany dysk i napisz

Save "example" [RETURN]

Po chwili ukaże się napis Ready; oznacza to, że program został zapisany na dysk. (Jeżeli tak się nie stanie - to przyjrzyj się komunikatowi o błędzie i sprawdź, czy przypadkiem nie zapomniałe włożyć dysku do właściwego napędu, zamknąć otworu zabezpieczającego przed zapisem lub też nie pomyliłeś się wpisując komendę.

Katalog (Skorowidz)

Po zapamiętaniu powyższego programu na dysku napisz:

cat [RETURN]

Na ekranie zobaczysz:

Drive A: user Ø

EXAMPLE.BAS 1K

177K free

Zostanie wyświetlona nazwa zbioru wraz z podanym bądź przyporządkowanym drugim polem nazwy oraz długość zbioru (zaokrąglona do najbliższej wielokrotności 1 KB, większej niż sama długość zbioru). Zostanie również podana wolna pojemność dysku.

Ładowanie z dysku do pamięci

Programy mogą być ładowane z dysku a następnie wykonywane przy użyciu komend:

load "nazwa zbioru" [RETURN]

run [RETURN]

... lub też mogą być bezpośrednio wykonywane po załadowaniu komendą

run "nazwa zbioru" [RETURN]

Należy zwrócić uwagę, że zbiory zabezpieczone mogą być uruchamiane wyłącznie w ten drugi sposób.

|A i |B

Jeżeli pracujesz z systemem o 2 napędach dyskowych, to możesz wybrać napęd w którym ma się dokonać dana operacja przez napisanie przed komendami SAVE, LOAD lub CAT komendy

|a [RETURN]

lub

|b [RETURN]

Kopiowanie programów z dysku na dysk

Widać już, że używając wyłącznie poznanych dotąd komend można kopiować programy z dysku na dysk. Wystarczy załadować program do pamięci z oryginalnego, źródłowego dysku, zamienić dyski i zapisać program na nowy, docelowy dysk.

Aby przepisać program z dysku na dysk w systemie z 2 napędami wystarczy umieścić dysk źródłowy w np. napędzie B: a dysk docelowy w napędzie A i napisać:

|b [RETURN]
load "nazwazbioru" [RETURN]
|a [RETURN]
save "nazwazbioru" [RETURN]

W CPC 6128 istnieją cztery metody zapisywania zbiorów na dysk. Obok zwykłego zapisywania zbiorów języka BASIC komendą:

save "nazwazbioru" [RETURN]

są jeszcze trzy alternatywne metody przeznaczone do bardziej specjalistycznych zastosowań:

Zbiory ASCII

save "nazwazbioru",a [RETURN]

Dodanie przyrostka, a zleca komputerowi zapisanie programu lub danych w postaci zbioru tekstowego w kodzie ASCII. Metodę tę stosuje się do zbiorów utworzonych przez procesory tekstów lub inne programy aplikacyjne, a szczegółowe omówienie tej metody znajdziemy w dalszej części podręcznika przy okazji napotkania jej zastosowań.

Zbiory zabezpieczone

save "nazwazbioru",p [RETURN]

Dodanie przyrostka, p nakazuje zabezpieczenie zbiorów w ten sposób, że programy takie nie mogą być wylistowane po załadowaniu oraz zatrzymane przez naciśnięcie klawisza [ESC] po uruchomieniu.

Programy zapisane w ten sposób mogą być uruchomione wyłącznie w sposób bezpośredni komendą

run "nazwazbioru" [RETURN]

lub

chain "nazwazbioru" [RETURN]

Jeżeli przewidujesz, że będziesz chciał dokonać edycji lub zmienić taki program, to koniecznie musisz posiadać jego kopię nie zabezpieczoną, zapisaną bez przyrostka ,p.

Zbiory binarne

save"nazwazbioru",b, <adres początkowy> , <długość w bajtach> ,[<opcjonalny adres startu>] [RETURN]

Ta opcja pozwala na zapisanie na dysk bloku danych z pamięci RAM w takiej postaci, w jakiej występuje w pamięci. Należy podać komputerowi, od jakiego adresu rozpoczyna się blok danych do zapisu, jaki jest długi i - jeżeli zachodzi potrzeba - adres, od którego ma się rozpocząć wykonywanie tak zapisanego programu po jego powtórnym załadowaniu do pamięci.

Kopia ekranu

Opcja zapisywania zbiorów binarnych pozwala na zapisanie na dysk pamięci ekranu, w postaci jego kopii. Zawartość ekranu zostanie zapisana tak, jak jest widziana przy pomocy komendy:

save "obrazekranu",b,49152,16384 [RETURN]

... gdzie 49152 jest początkowym adresem pamięci ekranu i 16384 ilością komórek pamięci, które trzeba zapisać.

Aby odtworzyć obraz na ekranie napisz:

load "obrazekranu" [RETURN]

Więcej informacji o tym, jak używać system do manipulowania zbiorami pomiędzy dyskami (i kasetami) znajdzie się w dalszej części podręcznika.

Na zakończenie upewnij się, że zastosowałeś się do następujących ważnych ostrzeżeń zawartych na początku tego podręcznika w rozdziale zatytułowanym "WAŻNE".

Uwagi o instalacji 5,6,7

Uwagi o obsłudze 1,2,3,4,5,6,7,9

Część 8: Tryby organizacji ekranu, kolory, grafika ...

Amstrad CPC6128 (Colour Personal Computer) ma trzy tryby (MODE) organizacji ekranu: tryb 0, tryb 1 i tryb 2.
Po włączeniu do sieci ustawiony jest tryb 1.

Włącz komputer i naciśnij klawisz 1. Naciskaj klawisz tak długo, aż zapełnią się jedynkami dwie linie. Jeżeli policzysz jedynki w linii, przekonasz się, że jest ich 40. Oznacza to, że w trybie 1 jest 40 kolumn na ekranie. Naciśnij [RETURN] - zobaczysz komunikat "Syntax error", ale nie martw się, jest to szybki sposób odzyskania komunikatu "Ready", który oznacza, że komputer czeka na następną instrukcję.
Wpisz teraz:

mode 0 [RETURN]

Litery na ekranie są teraz większe. Wciśnij znowu klawisz 1 i trzymaj tak długo jak poprzednio. Jeżeli policzysz jedynki w linii zobaczysz, że jest ich 20. Oznacza to, że w trybie 0 jest 20 kolumn. Naciśnij [RETURN] jeszcze raz.
Wpisz:

mode 2 [RETURN]

Znaki są teraz najmniejsze i, jeżeli wpiszesz całą linię znaków, możesz przekonać się, że jest ich 80.
Rekapitulując:
tryb 0 (mode 0) = 20 kolumn
tryb 1 (mode 1) = 40 kolumn
tryb 2 (mode 2) = 80 kolumn
Na koniec naciśnij [RETURN] jeszcze raz.

Kolory
Różnych kolorów jest 27. Na monitorze zielonym (GT65) są one różnymi odcieniami zieleni. Jeżeli kupiłeś monitor GT65 możesz dokupić Modulator/Zasilacz AMSTRAD MP2 co umożliwi ci dołączenie komputera do twojego domowego telewizora kolorowego i wykorzystanie możliwości tworzenia barwnej grafiki.
W trybie 0 na ekranie może być jednocześnie do 16 (z 27) kolorów.
W trybie 1 na ekranie może być jednocześnie do 4 (z 27) kolorów.
W trybie 2 na ekranie mogą być jednocześnie tylko 2 (z 27) kolory.

Można zmieniać kolor obrzeża (BORDER), papieru (PAPER) (powierzchnia, na której mogą pojawiać się znaki) lub pióra (PEN) (samych znaków) niezależnie jedno od drugiego.

W tablicy 1 zamieszczono wykaz kolorów dostępnych w CPC6128 wraz z ich numerami.

Dla wygody wykaz ten zamieszczono również na pokrywie komputera z prawej strony.

GŁÓWNA TABELA KOLORÓW

Nr	Kolor	Nr	Kolor	Nr	Kolor
0	Czarny	9	Zielony	18	J.zielony
1	Niebieski	10	Lazurowy	19	Zieleń morska
2	J.niebieski	11	Błękitny	20	J.lazurowy
3	Czerwony	12	Żółty	21	Cytrynowozielony
4	Szkarłatny	13	Biały	22	P.zielony
5	Bladofioletowy	14	P.niebieski	23	P.lazurowy
6	J.czerwony	15	Pomarańczowy	24	J.żółty
7	Purpurowy	16	Różowy	25	P.żółty
8	J.szkarłatny	17	P.szkarłatny	26	J.biały

J. - Jaskrawy; P. - Pastelowy

Tablica 1. Kolory i ich numery

Jak już wspomniano wcześniej, po włączeniu do sieci, komputer jest w trybie 1. Aby powrócić do trybu 1 z innego trybu, należy wpisać

```
mode 1 RETURN
```

Wyświetlanie na ekranie

Obrzeże (BORDER) jest to powierzchnia otaczająca papier (PAPER). (Zauważ, że po włączeniu do sieci zarówno obrzeże, jak i papier są koloru niebieskiego). Znaki na ekranie mogą pojawiać się tylko wewnątrz obrzeża. Papier jest tłem dla znaków, a pióro (PEN) pisze znaki.

Teraz wytłumaczymy, jaki jest mechanizm wyboru kolorów i jak możesz je zmieniać.

Po włączeniu do sieci lub wyzerowaniu komputera, obrzeże ma zawsze kolor nr 1. Spójrz na główną tabelę kolorów i zobacz

że kolor nr 1 jest niebieski. Kolor obrzeża może być zmieniony rozkazem BORDER, za którym podany jest numer koloru. Jeżeli chcesz zmienić kolor obrzeża na biały, wpisz:

border 13 [RETURN]

Jak dotąd, łatwo. Teraz będzie trudniej ...
Po włączeniu do sieci lub wyzerowaniu komputera, do papieru przypisany jest atrament o numerze 0, a do pióra atrament o numerze 1 To nie oznacza, że papier ma kolor 0, a pióro kolor 1 według głównej tabeli kolorów ...

Ważną rzeczą jest pamiętać, że 0 i 1 są w tym wypadku numerami atramentów, a nie numerami kolorów. Aby to zrozumieć, wyobraź sobie, że masz 4 buteleczki na atrament, o numerach 0, 1, 2 i 3 i atrament w każdej butelce możesz zabarwić jednym z 27 kolorów o numerach 0 do 26. Z tego wynika, że atrament o numerze 1 niekoniecznie musi być koloru o tym samym numerze, ponieważ może być zabarwiony innym kolorem. Możesz nawet zabarwić wszystkie atramenty tym samym kolorem.

Tak właśnie jest w komputerze. Rozkazy INK i PEN pozwalają kolejno zabarwić dany atrament kolorem i "napełnić" pióro danym atramentem.

Pamiętając, że mamy ustawiony tryb 1 (40 kolumn), spójrz na tablicę poniżej. Z pierwszej i trzeciej kolumny możesz odczytać, że atrament o numerze 1 (przypisany do pióra po wyzerowaniu komputera) zabarwiony jest kolorem 24. Spójrz teraz na główną tabelę kolorów (tablica 1), z której wynika, że jest to kolor jaskrawo żółty - taki właśnie kolor mają znaki na ekranie po wyzerowaniu.

POCZĄTKOWE ZABARWIENIE ATRAMENTÓW

Nr atramentu	Nr koloru Tryb 0	Nr koloru Tryb 1	Nr koloru Tryb 2
0	1	1	1
1	24	24	24
2	20	20	1
3	6	6	24
4	26	1	1
5	0	24	24
6	2	20	1
7	8	6	24
8	10	1	1
9	12	24	24
10	14	20	1
11	16	6	24
12	18	1	1
13	22	24	24
14	Przełączane 1,24	20	1
15	Przełączane 16,11	6	24

Tablica 2. Zależności atrament /tryb /kolor

Zależności pomiędzy atramentem, trybem i kolorem podane w tablicy 2 nie są niezmienne. Jest to ustalenie wstępne, przyjmowane przy wyzerowaniu lub włączeniu do sieci komputera. Zmianę zabarwienia atramentu umożliwia rozkaz INK (atrament). Rozkaz ten ma dwa parametry. Pierwszym parametrem jest numer atramentu, który ma być zabarwiony, drugim jest numer koloru. Parametry te muszą być oddzielone przecinkiem.

Tak więc, wiedząc, że w piórze jest atrament o numerze 1, zmieńmy kolor tego atramentu na pomarańczowy.

Wpisz:

ink 1,15 [RETURN]

..., a zobaczysz, że litery na ekranie zmieniły kolor.

Kolor tła może być również zmieniony rozkazem INK. Wiemy, że po włączeniu komputera, do papieru przypisany jest atrament 0. Zmienmy więc barwę tego atramentu na zieloną (kolor nr 9) wpisując:

ink 0,9 [RETURN]

Przypiszmy teraz do pióra inny atrament. Wpisz:

pen 3 [RETURN]

Zauważ, że zmienia się kolor tylko nowych liter (po rozkazie). Używasz teraz do pisania atramentu nr 3 i z tablicy 1 i 2 możesz znaleźć, że kolor atramentu nr 3 jest jaskrawo czerwony (nr 6). Zmień go na różowy wpisując:

ink 3,16 [RETURN]

Pamiętaj, że 3 jest numerem atramentu, który wybrałeś do pisania rozkazem PEN 3, a 16 jest numerem koloru różowego. Zmieńmy teraz atrament przypisany do papieru. Po wybraniu nowego atramentu poprzednie tło pod znakami NIE zmienia się, ponieważ było ono "drukowane" innym atramentem. Aby to zobaczyć wpisz:

paper 2 [RETURN]

Znowu skorzystaj z tablic 1 i 2, żeby stwierdzić, że kolor dla atramentu nr 2 jest jaskrawo zielono niebieski. Zmień go na czarny:

ink 2,0 [RETURN]

Teraz, na ekranie mamy znaki wpisane piórem, atramentami numer 1 i 3 i tła "namalowane" atramentami numer 0 i 2. Barwa atramentu może być zmieniana również w przypadku, gdy ten atrament nie jest aktualnie przypisany ani do papieru, ani do pióra. Na przykład, wpisz:

ink 1,2 [RETURN]

... co zmienia kolor wszystkich znaków wpisanych wcześniej piórem z atramentem numer 1.
Wpisz:

cls [RETURN]

..., aby wyczyścić ekran.

Powinieneś teraz potrafić poinstruować komputer jak wrócić do kolorów początkowych (niebieskie obrzeże i tło oraz jaskrawo żółte litery). Spróbuj to zrobić rozkazami BORDER, PAPER, PEN i INK. Jeżeli ci się nie uda, po prostu wyzeruj komputer przez naciśnięcie klawiszów [CONTROL] [SHIFT] i [ESC].

Kolory przełączane

Można spowodować, że kolor znaków jest przełączany pomiędzy dwoma kolorami. W tym celu w rozkazie INK barwiącym atrament przypisany do pióra należy dodać trzeci parametr, który jest numerem drugiego koloru. Aby zobaczyć na ekranie litery, których kolor zmienia się z jaskrawo białego na jaskrawo czerwony, wyzeruj komputer ([CONTROL][SHIFT][ESC]) i wpisz:

ink 1,26,6 [RETURN]

Tutaj 1 jest numerem atramentu przypisanego do pióra, 26 jest kolorem jaskrawo białym, a 6 jest kolorem jaskrawo czerwonym. Również tło może mieć kolor przełączany. W tym celu w rozkazie INK, barwiącym atrament przypisany do papieru, należy dodać numer drugiego koloru.

Aby zobaczyć na ekranie tło, którego kolor zmienia się z zielonego na jaskrawo żółte, wpisz:

ink 0,9,24 [RETURN]

Tutaj 0 jest numerem atramentu przypisanego do papieru, 9 oznacza kolor zielony, 24 - kolor jaskrawo żółty.
Wyzeruj teraz komputer ([CONTROL][SHIFT][ESC]).

Zobacz w tablicy 2, że w trybie 0, dwa atramenty (numery 14 i 15) są wstępnie zabarwione dwukolorowo. Innymi słowy w rozkazach INK barwiących te atramenty został dodany trzeci parametr
Wpisz poniższe:

mode 0 [RETURN]
pen 15 [RETURN]

zobaczysz, że słowo "Ready" jest na zmianę błękitne i różowe.
Teraz wpisz:

paper 14 [RETURN]
cls [RETURN]

Zobaczysz, że prócz przełączania barw liter "Ready" pomiędzy kolorem błękitnym i różowym następuje również przełączanie barw tła z żółtego na niebieskie.

Atramenty 14 i 15 mogą być przeprogramowane rozkazem INK na inne przełączane kolory lub na kolory stałe.

W końcu można również przełączać kolor obrzeża, co można uzyskać przez dodanie drugiego koloru w rozkazie BORDER.
Wpisz:

border 6,9 [RETURN]

Zobaczysz, że kolor obrzeża jest przełączany z jaskrawo czerwonego na zielony. Zwróć uwagę na to, że kolor obrzeża może być ustalony na którykolwiek (lub którąkolwiek parę kolorów) spośród 27 kolorów, bez względu na aktualny tryb.

Wyzeruj **komputer**
W celu demonstracji dostępności kolorów wpisz, a następnie uruchom następujący program:

```
10 MODE 0 [RETURN]
20 czas=600:REM ustala szybkość zmian [RETURN]
30 FOR b=0 TO 26 [RETURN]
40 LOCATE 3,12 [RETURN]
50 BORDER b [RETURN]
60 PRINT "kolor obrzeża";b[RETURN]
70 FOR t=1 TO czas [RETURN]
80 NEXT t,b [RETURN]
90 CLG [RETURN]
100 FOR p=0 TO 15 [RETURN]
110 PAPER p [RETURN]
120 PRINT "papier"; p [RETURN]
130 FOR n=0 TO 15 [RETURN]
140 PEN n [RETURN]
150 PRINT "pióro"; n [RETURN]
160 NEXT n [RETURN]
170 FOR t=1 TO czas*2 [RETURN]
180 NEXT t,p [RETURN]
190 MODE 1 [RETURN]
200 BORDER 1 [RETURN]
```

```
210  PAPER  0 [RETURN]
220  PEN 1 [RETURN]
230  INK 0,1 [RETURN]
240  INK 1,24 [RETURN]
run [RETURN]
```

Ważne

W powyższym programie, jak również w programach podanych w następnych rozdziałach, słowa kluczowe BASIC'a napisane są dużymi (CAPital) literami. W ten sposób słowa te są pisane przez BAISC przy listowaniu. Ogólnie rzecz biorąc, leniej jest używać przy pisaniu programu liter małych, ponieważ ułatwia to znalezienie błędów wpisywania (błędnie wpisane słowa kluczowe przy listowaniu NIE będą napisane dużymi literami).

W pozostałej części niniejszego rozdziału używamy zarówno dużych jak i małych liter, aby przyzwyczaić cię do tego sposobu działania.

Nazwy zmiennych (np. x lub a$) NIE są zamieniane na duże litery przy listowaniu, chociaż komputer rozpoznaje nazwę zmiennej bez względu na to, czy jest ona napisana dużymi czy małymi literami.

Uwaga

Od tej chwili nie będziemy pisać [RETURN] na końcu każdej linii. Zakładamy, że będziesz to robić automatycznie.

Grafika

W pamięci komputera zapamiętane są pewne znaki symboliczne. W celu wydrukowania któregokolwiek z nich należy użyć słowa:

chr$( )

Wewnątrz nawiasów powinien być numer znaku z zakresu 32 do 255. Wyzeruj komputer ([CONTROL] [SHIFT] [ESC]) i wpisz:

print chr$ (250)

Nie zapomnij nacisnąć [RETURN]. Na ekranie pojawi się znak numer 250, którym jest człowiek idący w prawo.

Jeżeli chcesz zobaczyć wszystkie znaki i symbole, które mogą być drukowane na ekranie oraz ich numery, wpisz następujący program (przypominamy o naciśnięciu [RETURN] na końcu każdej linii).

```
10 for n=32 to 255
20 print n; chr$(n)
30 next n
run
```

Wszystkie znaki oraz ich numery podano w rozdziale zatytułowanym "Nieco użytecznych informacji ..."

LOCATE

Rozkaz ten jest używany w celu przemieszczenia kursora tekstowego na określone miejsce na ekranie. Jeżeli pozycja kursora tekstowego nie jest zmieniona rozkazem LOCATE, drukowanie tekstu zaczyna się od lewego, górnego rogu ekranu, co odpowiada współrzędnym (x,y) = (1,1) (x jest pozycją poziomą, y jest pozycją pionową). W trybie 1 kolumn jest 40, a linii 25. Dlatego, aby umieścić znak w środku górnej linii w trybie 1, należy określić współrzędne x,y jako 20,1.

Aby to zobaczyć, wpisz (pamiętaj o [RETURN] na końcu każdej linii):

```
10 locate 20,1
20 print chr$(250)
run
```

Aby się przekonać, że jest to najwyższa linia, wpisz:

```
border 0
```

Obrzeże jest teraz czarne, a człowieczek znajduje się w środku najwyższej linii.

W trybie 0 kolumn jest tylko 20, linii, tak samo jak poprzednio, 25. Jeżeli teraz wpiszesz:

```
mode 0
run
```

..., zobaczysz, że człowieczek jest teraz w prawym, górnym rogu ekranu. Jest tak, ponieważ współrzędna x równa 20 określa ostatnią kolumnę w trybie 0.

W trybie 2 jest 80 kolumn i 25 linii. Prawdopodobnie potrafisz zgadnąć, w którym miejscu ekranu pojawi się człowieczek po wpisaniu:

```
mode 2
run
```

Wróć do trybu 1 wpisując:

```
mode 1
```

Sam wypróbuj, przez modyfikację parametrów instrukcji LOCATE i CHR$, rozmieszczenie różnych znaków na ekranie. Na przykład wpisz:

```
locate 20,12:print chr$(240)
```

Zobaczysz strzałkę w środku ekranu. Zauważ, że w tej instrukcji:
20 jest współrzędną poziomą (x) (w zakresie 1 do 40)
12 jest współrzędną pionową (y) (w zakresie 1 do 25)
240 jest numerem znaku symbolicznego (w zakresie 32 do 255).
Aby otrzymać znak numer 250, powtarzany przez cały ekran wpisz:

```
10 CLS
20 FOR x=1 TO 39
30 LOCATE x,20
50 PRINT CHR$(250)
60 NEXT X
70 GOTO 10
run
```

Naciśnij dwukrotnie klawisz [ESC], aby przerwać działanie programu.

Chcąc usunąć wcześniej wpisany znak przed wydrukowaniem następnego (aby był tylko jeden człowiek, a nie cała linia) wpisz:

```
50 print " "; chr$(250)
```

(Nowa linia 50 automatycznie zastępuje poprzednią)
Teraz wpisz:

```
run
```

FRAME

Aby ulepszyć sposób poruszania się znaku na ekranie dodaj następującą linię:

```
40 frame
```

Komenda FRAME synchronizuje przesuwanie się obiektów na ekranie z częstotliwością ramki obrazu monitora. Jeżeli brzmi to dla Ciebie zbyt technicznie, to zapamiętaj tylko, że komenda ta powinna być użyta zawsze, kiedy pragniesz, aby tekst lub grafika płynnie przesuwały się na ekranie.

Program może być dalej rozszerzony w celu poprawienia jakości ruchu poprzez dodanie kilku pętli opóźniających i użycie innego symbolu znaku.

Napisz

```
list
```

Teraz dodaj następujące linijki programu:

```
 70 FOR n=1 TO 3ØØ:NEXT n
 80 FOR x=39 TO 1 STEP-1
 90 LOCATE x,2Ø
100 FRAME
110 PRINT CHR$(251);" "
120 NEXT x
130 FOR n=1 TO 300:NEXT n
140 GOTO 20
run
```

PLOT

Komenda PLOT, co odróżnia ją od LOCATE, może być użyta do określenia pozycji kursora graficznego we współrzędnych pojedynczych punktów (ang. "pixels"). Punkt jest najmniejszym osiągalnym elementem ekranu.

Zwróćmy uwagę, że kursor graficzny nie jest widoczny i różni się od kursora znakowego (tekstowego).

Na ekranie może być rozróżnione 640 punktów w poziomie i 400 w pionie. Współrzędne x i y określone są w stosunku do lewego dolnego rogu ekranu. Ma on współrzędne 0,0. W przeciwieństwie do kursora znakowego - współrzędne punktu nie różnią się w trybach (modach) Ø, 1 i 2.

Aby się o tym przekonać wyresetuj komputer naciskając przyciski [CONTROL][SHIFT] i [ESC] a następnie napisz

```
plot 32Ø,2ØØ
```

Na środku ekranu pojawi się mały punkt. Zmieńmy teraz tryb pisząc:

```
mode Ø
plot 32Ø, 2ØØ
```

Zobaczysz, że punkt na środku ekranu jest dalej - jest on jednak większy.

Zmień tryb pracy powtórnie i napisz tę samą komendę aby zobaczyć efekt pracy w trybie 2:

```
mode 2
plot 320,200
```

Punkt jest w dalszym ciągu, tym razem jednak znacznie mniejszy.

Wpisz w ekran punkty w różnych trybach i miejscach. Da ci to pewną wprawę w używaniu komendy. Kiedy już skończysz - powróć do trybu 1 i oczyść ekran pisząc:

```
mode 1
```

DRAW

Wpierw wyzeruj komputer używając klawiszy [CONTROL][SHIFT] [ESC]. Komenda DRAW kreśli na ekranie linię wychodzącą z punktu, w którym usytuowany jest kursor graficzny. Aby poznać to w szczegółach, przy pomocy poniższego programu narysuj na ekranie trójkąt.

Zaczniemy przesuwając kursor graficzny komendą PLOT a następnie kreśląc (komendą DRAW) linię z pozycji kursora graficznego do góry w kierunku lewego górnego rogu, następnie prawego rogu itd.

Napisz:

```
 5 cls
1Ø plot 1Ø,1Ø
2Ø draw 1Ø,39Ø
3Ø draw 63Ø,39Ø
4Ø draw 63Ø,1Ø
5Ø draw 1Ø,1Ø
6Ø goto 6Ø
run
```

Aby zatrzymać ten program naciśnij dwa razy [ESC]. (Zauważ, że w linii 60 komputer został poinstruowany, aby działał "w pętli" tak długo, dopóki nie zostanie zatrzymany dwukrotnym naciśnięciem [ESC]. Taka instrukcja jest użyteczna, jeżeli nie chcesz, aby program samoistnie zatrzymał się na końcu i wyświetlił napis "Ready", który zwykle towarzyszy zakończeniu programu).

Teraz dodaj następujące linie rysujące drugi trójkąt wewnątrz pierwszego:

```
60 plot 20,20
70 draw 20,380
80 draw 620,380
90 draw 620,20
100 draw 20,20
110 goto 110
run
```

I znów, aby przerwać program naciśnij dwa razy [ESC].

MOVE

Komenda MOVE działa w sposób podobny do PLOT, tj. kursor graficzny przesuwany jest do miejsca o współrzędnych x i y z tym jednak, że w nowym miejscu ustawienia kursora nie jest stawiany punkt (kropka).

Napisz:

```
cls
move 639,399
```

Chociaż efektów tej komendy nie widać na ekranie, to przesunęliśmy kursor graficzny do górnego prawego rogu ekranu.

Przekonajmy się o tym, kreśląc linię z aktualnego miejsca usytuowania kursora graficznego do środka ekranu:

```
draw 320,200
```

Okręgi

Okręgi mogą być kreślone z odcinków prostych lub tworzone z punktów. Jedną z metod jest rysowanie okręgu poprzez stawianie punktów na jego obwodzie.

Jak widać z rysunku Punkt p na obwodzie może być postawiony używając współrzędnych x i y. Są one wyznaczane jako:

x = 19Ø*cos(a)

y = 19Ø*sin(a)

Rysowanie okręgu z punktów

W poprzednim programie określaliśmy punkty w stosunku do lewego dolnego rogu ekranu. Jeżeli chcemy umieścić okrąg na środku ekranu, to musimy umieścić środek okręgu we współrzędnych 32Ø, 2ØØ a następnie odnieść wszystkie punkty okręgu do jego środka poprzez dodanie jego współrzędnych.

Program rysowania okręgu byłby taki:

```
new
1Ø CLS
2Ø DEG
3Ø FOR a=1 TO 36Ø
4Ø MOVE 32Ø,2ØØ
5Ø PLOT 32Ø+19Ø*COS(a),2ØØ+19Ø*SIN(a)
6Ø NEXT
run
```

Zwróćmy uwagę na użycie słowa kluczowego n e w wykorzystanego przed wpisaniem programu. Informuje ono komputer, że należy zlikwidować w pamięci poprzedni program (w sposób podobny do [CONTROL][SHIFT] i [ESC]). Sama zawartość ekranu nie zostaje naruszona.

Promień okręgu może być zmniejszony przez zastąpienie liczby 190 dowolną mniejszą liczbą (190 odpowiada ilości punktów)

Aby zobaczyć efekt innego kreślenia okręgu (w radianach), zlikwidujmy linię 2Ø pisząc

. 2Ø

Aby zobaczyć wypełniony okrąg rysowany liniami wychodzącymi z jego środka, dokonaj edycji linii 5Ø zastępując słowo plot słowem draw. (Linia 5Ø będzie więc mieć postać):

```
5Ø draw 32Ø+19Ø*cos(a),2ØØ+190*sin(a)
```

Spróbuj narysować taki okrąg raz z, a następnie bez linii 2Ø.

Zauważ, że w linii 60 tego programu użyliśmy NEXT zamiast NEXT a. Wolno jest pisać samo NEXT; komputer sam znajdzie odpowiednią komendę FOR. Jednakże, w programach, w których występują liczne pętle FOR i NEXT, warto czasem dodać nazwę zmiennej po NEXT - ułatwia to czytanie programu i określenie zasięgu każdej pętli.

ORIGIN

W poprzednim programie użyliśmy komendy MOVE w celu określenia środka okręgu, a następnie dodaliśmy współrzędne x i y do pozycji środka. Zamiast dodawać współrzędne środka możemy użyć komendy ORIGIN. Sprawi to, że każde współrzędne (x,y) będą odniesione do pozycji określonej komendą ORIGIN. Aby się o tym przekonać, napisz:

```
new
1Ø cls
2Ø for a=1 to 36Ø
3Ø origin 32Ø,2ØØ
4Ø plot 19Ø*cos(a),190*sin(a)
5Ø next
run
```

W celu wykreślenia mniejszych okręgów na ekranie wprowadź następujący program:

```
new
1Ø CLS
2Ø FOR a=1 TO 36Ø
3Ø ORIGIN 196,282
4Ø PLOT 5Ø*cos(a),5Ø*sin(a)
5Ø ORIGIN 442,282
6Ø PLOT 5Ø*cos(a),5Ø*sin(a)
7Ø ORIGIN 196,116
8Ø PLOT 5Ø*cos(a),5Ø*sin(a)
9Ø ORIGIN 449,116
1ØØ PLOT 5Ø*cos(a),5Ø*sin(a)
11Ø NEXT
run
```

Inny sposób tworzenia okręgu przedstawia program:

```
new
10 mode 1
20 origin 320,300
30 deg
40 move 0,190
50 for a=0 to 360 step 10
60 draw 190*sin(a),190*cos(a)
70 next
run
```

Tym razem pomiędzy współrzędnymi na obwodzie okręgu kreślone są linie. Zwróć uwagę, że okrąg kreślony jest znacznie szybciej.

Jeszcze raz zaobserwuj efekt wyeliminowania komendy DEG; usuń linię 30 i napisz run.

FILL

Komenda FILL używana jest do wypełniania kolorem obszaru obramowanego wykreślonymi liniami, brzegami ekranu lub okienka graficznego.

Wyzeruj komputer [CONTROL][SHIFT][ESC] a następnie napisz:

```
new
10 cls
20 move 20,20
30 draw 620,20
40 draw 310,380
50 draw220,20
run
```

Na ekranie pojawi się trójkąt. Naprowadź kursor graficzny na środek trójkąta pisząc:

```
move 320,200
```

Używając komendy FILL z argumentem w postaci numeru atramentu, wypełnimy teraz (ang. "fill") obszar od pozycji kursora do najbliższego obramowania.

Napisz:

```
fill 3
```

Przesuń teraz kursor poza trójkąt pisząc:

```
move Ø,Ø
```

Zobacz co się stanie po napisaniu:

```
fill 2
```

Komputer użył atramentu numer 2 do wypełnienia obszaru ograniczonego narysowanymi liniami i brzegiem ekranu.

Zmień teraz program wpisując następujące linijki i zobacz co się stanie:

```
5Ø draw 5Ø,5Ø
6Ø move 32Ø,2ØØ
7Ø fill 3
run
```

Zobaczysz, że przez najmniejszą przerwę w obramowaniu atrament "wypływa" na zewnątrz.

Zjawisko to możemy dalej zademonstrować wypełniając wpierw okrąg, utworzony z punktów, a następnie wykreślony. Napisz:

```
new
1Ø CLS
2Ø FOR a=1 TO 36Ø
3Ø ORIGIN 32Ø,2ØØ
4Ø PLOT 19Ø*COS(a),19Ø*SIN(a)
5Ø NEXT
6Ø MOVE - 188,Ø
7Ø FILL 3
run
```

Teraz spróbuj tak:

```
new
1Ø MODE 1
2Ø ORIGIN 32Ø,2ØØ
3Ø DEG
4Ø MOVE Ø,19Ø
5Ø FOR d=Ø TO 36Ø step 1Ø
6Ø DRAW 19Ø*SIN(d), 19Ø*COS(d)
7Ø NEXT
8Ø MOVE - 188,Ø
9Ø FILL 3
run
```

Możemy również uczynić okrąg niewidocznym; należy ustawić atrament pióra na taki sam kolor jak tło. Dodaj:

```
45 GRAPHICS PEN 2:INK 2,1
run
```

Komenda GRAPHICS PEN wybiera atrament, który ma być używany do tworzenia grafiki na ekranie. Kolor temu atramentowi nadaje komenda INK, która w tym wypadku wybiera kolor tła (tj. kolor numer 1).

Na koniec wpisz następujący program demonstracyjny.

```
new
10 MODE 0:BORDER 13
20 MOVE 0,200:DRAW 640,200
30 FOR x=80 TO 560 STEP 80
40 MOVE x,0:DRAW x,400
50 NEXT:MOVE -40,300
60 FOR c=0 TO 7
70 MOVER 80,0:FILL C
80 MOVER 0,-200:FILL C+8
90 MOVER 0,200:NEXT
100 GOTO 100
run
```

Kolory pól mogą być po wypełnieniu zmieniane.

Napisz:

```
100 SPEED INK 30,30
110 BORDER RND*26, RND*26
120 INK RND*15, RND*26, RND*26
130 FOR t=1 TO 500:NEXT:GOTO 110
run
```

Dalsze szczegóły

Po bardziej zrozumiałe informacje na temat grafiki CPC 6128 sięgnij do rozdziału zatytułowanego "W wolnej chwili ..."

Aby zakończyć ten rozdział podajemy kilka programów demonstracyjnych używających wielu komend i słów kluczowych, które powinny być już zrozumiałe. Wszystkie te programy nieprzerwanie rysują na ekranie pewne wzory.

```
new
10 BORDER 0:GRAPHICS PEN 1
20 m=CINT(RND*2):MODE m
30 i1=RND*26:i2=RND*26
40 IF ABS(i1-i2)<10 THEN 30
50 INK 0,i1:INK 1,i2
60 s=RND*5+3:ORIGIN 320,-100
70 FOR x= -1000 TO 0 STEP s
80 MOVE 0,0:DRAW x,300:DRAW 0,600
90 MOVE 0,0:DRAW -x,300:DRAW 0,600
100 NEXT:FOR t=1 TO 2000:NEXT:GOTO 20
run
```

```
10 MODE 1:BORDER 0:PAPER 0
20 GRAPHICS PEN 2:INK 0,0:i=14
30 EVERY 2200 GOSUB 150
40 flag=0:CLG
50 INK 2,14+RND*12
60 b%=RND*5+1
70 c%=RND*5+1
80 ORIGIN 320,200
90 FOR a=0 TO 1000 STEP PI/30
100 x%=100*COS(a)
110 MOVE x%,y%
120 DRAW 200*COS(a/b%),200*SIN(a/c%)
130 IF flag=1 THEN 40
140 NEXT
150 flag=1:RETURN
run
```

```
10 MODE 1:BORDER 0:DEG
20 PRINT "Please wait"
30 FOR n=1 TO 3
40 INK 0,0:INK 1,26:INK 2,6:INK 3,18
50 IF n=1 THEN sa=120
60 IF n=2 THEN sa=135
70 IF n=3 THEN sa=150
80 IF n=1 THEN ORIGIN 0,-50,0,640,0,400 ELSE ORIGIN 0,0
   ,0,640,0,400
90 DIM cx(5),cy(5),r(5),lc(5)
100 DIM np(5)
110 DIM px%(5,81),py%(5,81)
120 st=1:cx(1)=320:cy(1)=200:r(1)=80
130 FOR st=1 TO 4
140 r(st+1)=r(st)/2
150 NEXT st
160 FOR st=1 TO 5
170 lc(st)=0:np(st)=0
180 np(st)=np(st)+1
190 px%(st,np(st))=r(st)*SIN(lc(st))
200 py%(st,np(st))=r(st)*COS(lc(st))
210 lc(st)=lc(st)+360/r(st)
220 IF lc(st) < 360 THEN 180
230 px%(st,np(st)+1)=px%(st,1)
240 py%(st,np(st)+1)=py%(st,1)
250 NEXT st
260 CLS:cj=REMAIN(1):cj=REMAIN(2)
270 cj=REMAIN(3):INK 1,2:st=1
280 GOSUB 350
290 LOCATE 1,1
300 EVERY 25,1 GOSUB 510
310 EVERY 15,2 GOSUB 550
320 EVERY 5,3 GOSUB 590
330 ERASE cx,cy,r,lc,np,px%,py%:NEXT
340 GOTO 340
350 cx%=cx(st):cy%=cy(st):lc(st)=0
360 FOR x%=1 TO np(st)
370 MOVE cx%,cy%
380 DRAW cx%+px%(st,x%),cy%+py%(st,x%),1+(st MOD 3)
390 DRAW cx%+px%(st,x%+1),cy%+py%(st,x%+1),1+(st MOD 3)
400 NEXT x%
410 IF st=5 THEN RETURN
420 lc(st)=0
430 cx(st+1)=cx(st)+1.5*r(st)*SIN(sa+lc(st))
```

```
440 cy(st+1)=cy(st)+1.5*r(st)*COS(sa+lc(st))
450 st=st+1
460 GOSUB 350
470 st=st-1
480 lc(st)=lc(st)+2*sa
490 IF (lc(st) MOD 360)<>0 THEN 430
500 RETURN
510 ik(1)=1+RND*25
520 IF ik(1)=ik(2) OR ik(1)=ik(3) THEN 510
530 INK 1,ik(1)
540 RETURN
550 ik(2)=1+RND*25
560 IF ik(2)=ik(1) OR ik(2)=ik(3) THEN 550
570 INK 2,ik(2)
580 RETURN
590 ik(3)=1+RND*25
600 IF ik(3)=ik(1) OR ik(3)=ik(2) THEN 590
610 INK 3,ik(3)
620 RETURN
```

Część 9. Używanie dźwięku

Dźwięk generowany jest przez głośnik wbudowany do komputera. Jeżeli używasz Modulatora/Zasilacza MP2 i domowego odbiornika TV to pokrętło głośności w telewizorze należy ustawić na minimum.

Poziom głośności może być wyregulowany pokrętłem VOLUME po prawej stronie komputera. Dźwięki generowane przez komputer mogą być również doprowadzone do dodatkowego (zewnętrznego) wejścia zestawu stereofonicznego przy użyciu wyjścia z komputera oznaczonego STEREO. Umożliwi to słuchanie dźwięków generowanych przez komputer w wersji stereofonicznej - albo za pośrednictwem głośników hi-fi, albo słuchawek. Instrukcja podłączenia urządzeń do złącza STEREO komputera znajduje się w części 2 tego rozdziału.

Komenda SOUND

Komenda SOUND ("Dźwięk") ma 7 części (zwanych parametrami) Dwa pierwsze parametry muszą być użyte; reszta jest opcjonalna. Komendę podaje się jako:

SOUND <status kanału>, <okres tonu>, <czas trwania>,
<głośność>, <obwiednia głośności>, <obwiednia tonu>,
<okres szumu>

Wygląda to całkiem skomplikowanie, ale jeżeli przeanalizujemy po kolei wszystkie parametry to wkrótce oswoimy się z nimi. Przypatrzmy się tym parametrom po kolei.

Status kanału

Dla uproszczenia, na początek potraktujmy status kanału jako odniesienie do danego kanału. Są trzy kanały dźwiękowe i na razie będziemy używać <status kanału> numer 1.

Okres tonu

Okres tonu jest technicznym sposobem definiowania wysokości dźwięku, czyli innymi słowy "jaka to nuta" (tj. do, re, mi, fa, sol itd.). Każda nuta ma swój numer i jest to właśnie

<okres tonu>. Zajrzyj do rozdziału zatytułowanego "Trochę użytecznych informacji", a przekonasz się, że nuta "środkowe c (do)" ma okres tonu równy 239.

Wyresetuj teraz komputer ; [CONTROL][SHIFT][ESC] i napisz:

```
10 sound 1,239
run
```

Usłyszysz krótki, trwający 0,2 s dźwięk c. Jeżeli nic nie usłyszałeś, to upewnij się, czy pokrętło VOLUME nie jest zupełnie skręcone i powtórnie uruchom program.

Czas trwania

Parametr ten ustala długość dźwięku, czyli innymi słowy "jak długo on trwa". Parametr operuje jednostkami 0,01 s (jedna setna) i jeżeli nie zostanie on podany to przyjmowana jest wartość 20. Wyjaśnia to, dlaczego dźwięk który dopiero co usłyszałeś trwał 0,2 sek (0,01 razy 20).

Aby dźwięk trwał 1 sek należy użyć czas trwania równy 100; dla 2 sek będzie to 200 itd. Napisz

```
10 sound 1,239,200
run
```

Usłyszysz środkowe c trwające 2 sek.

Głośność

Parametr ten określa początkową głośność dźwięku. Zakres tego parametru zmienia się od 0 do 15. <Głośność> równa 0 jest minimalna, 15 - maksymalna. Jeżeli parametr zostanie pominięty, to przyjmowana jest wielkość 12. Napisz:

```
10 sound 1,239,200,5
run
```

Zwróć uwagę na głośność dźwięku. Napisz to powtórnie, zwiększając ją:

```
10 sound 1,239,200,15
run
```

Zauważysz, że usłyszałeś znacznie głośniejszy dźwięk.

Obwiednia głośności

Aby głośność dźwięku zmieniała się w czasie jego trwania, można użyć osobnej komendy ENV. W rzeczywistości możesz utworzyć wiele różnych obwiedni głośności i, podobnie jak komenda SOUND, obwiednie będą mieć swoje numery. Jeżeli utworzyłeś obwiednię o numerze 1 i pragniesz użyć jej w komendzie SOUND, to należy w miejsce <obwiednia głośności> wpisać cyfrę 1. Tworzenie obwiedni głośności będzie wkrótce wyjaśnione.

Obwiednia tonu

Aby wysokość dźwięku zmieniała się w czasie jego trwania, możesz utworzyć obwiednię tonu osobnym rozkazem ENT. W rzeczywistości możesz utworzyć wiele obwiedni głośności i, podobnie jak komenda SOUND, każda z nich będzie miała swój numer. Jeżeli utworzysz obwiednię o numerze 1 i pragniesz użyć jej w komendzie SOUND, to w miejsce obwiednia tonu należy wpisać cyfrę 1. Tworzenie obwiedni tonu będzie wkrótce wyjaśnione.

Szum

<szum> jest ostatnim parametrem komendy SOUND. Zakres szumu zmienia się od 1 do 31. Dodaj parametr <szum> równy 2 na końcu komendy SOUND i oceń efekt. Następnie zmień wartość tego parametru na 27 i zwróć uwagę na różnicę. Napisz:

```
10 sound 1,239,200,15,,,2
```

Zwróć uwagę na dwa "puste" parametry (,,,) przed parametrem 2. Należy je wstawić, gdyż nie utworzyliśmy ani obwiedni głośności ani tonu.

Tworzenie obwiedni głośności

Dokonuje się tego za pomocą rozkazu ENV. W najprostszej postaci komenda ma 4 parametry. Wprowadza się je jako:

ENV <numer obwiedni>, <ilość kroków>, <rozmiar kroku>, <czas kroku>

Tak jak poprzednio, przyjrzyjmy się po kolei poszczególnym parametrom.

Numer obwiedni

Jest to liczba, dzięki której obwiednie mogą być rozróżniane w komendzie SOUND. Zakres: Ø do 15.

Ilość kroków

Parametr ten podaje, przez ile poziomów głośności ma przejść dźwięk przed swoim zakończeniem. Dla przykładu: jeżeli dźwięk ma trwać 10 sekund, to możesz pragnąć, aby było 10 kroków głośności; każdy trwający 1 sek. W takim wypadku <ilość kroków> należy ustawić na 10.
Dopuszczalny zakres tego parametru: Ø do 127.

Rozmiar kroku

Każdy krok może mieć głośność inną w stosunku do poprzedniego o 0 do 15 jednostek. Natężenia głośności są takie jak w komendzie SOUND z tym, że zakres tego parametru może się zmieniać od -128 do +127. Dzieje się tak dlatego, że po głośności 15 z powrotem nastąpi głośność Ø.

Czas kroku

Parametr ten określa czas pomiędzy krokami w jednostkach 0,01 sek. Zakres parametru: 0 do 255, co oznacza, że jeden krok może trwać do 2,56 sek. (Ø traktowane jest jako 256).
Należy zwrócić uwagę, że <ilość kroków> pomnożona przez <czas kroku> nie powinna być większa niż wartość parametru <czas trwania> w komendzie SOUND.

W przeciwnym razie dźwięk się skończy zanim przejdzie przez wszystkie kroki głośności. (W takim przypadku pozostała część obwiedni jest pomijana).

Podobnie, jeżeli parametr <czas trwania> w komendzie SOUND jest większy niż <ilość kroków> razy <czas kroku> to dźwięk będzie trwał po przejściu przez wszystkie stopnie obwiedni i zostanie utrzymany na poziomie ostatniego kroku.

W celu eksperymentu z rozkazem ENV wpisz program:

```
1Ø env 1,1Ø,1,1ØØ
2Ø sound 1,142,1ØØØ,1,1
run
```

Linia 2Ø określa dźwięk 142 (międzynarodowe a), trwający 10 sekund, o początkowej głośności 1 i obwiedni głośności numer 1. Obwiednia ta składa się z 10 kroków, zwiększających co 1 sek. (0,01 x 100) głośność o 1.

Zmień następnie linię 1Ø na następujące linie:

```
1Ø env 1,1ØØ,1,1Ø
1Ø env 1,1ØØ,2,1Ø
1Ø env 1,1ØØ,4,1Ø
1Ø env 1,5Ø,2Ø,2Ø
1Ø env 1,5Ø,2,2Ø
1Ø env 1,5Ø,15,3Ø
```

W końcu spróbuj tak:

```
10 env 1,5Ø,2,1Ø
```

Zauważysz, że od połowy dźwięku jego poziom głośności pozostaje stały. Jest tak, ponieważ ilość kroków określona była na 5Ø a czas kroku na 0,1 sek. Zatem czas przejścia przez obwiednię był określony na 5 sek a czas trwania dźwięku na 1Ø (gdyż parametr <czas trwania> wynosił 1ØØØ).

Teraz samodzielnie utwórz kilka obwiedni i zobacz jakie uzyskasz efekty.

Jeżeli pragniesz utworzyć bardziej interesujące obwiednie, to na końcu komendy ENV parametry ilość kroków , rozmiar kroku i czas kroku mogą być powtórzone jeszcze do 4 razy, tworząc różne "sekcje" jednej obwiedni.

Tworzenie obwiedni tonu

Tworzenie obwiedni tonu dokonuje się za pomocą komendy ENT. W swej najprostszej formie ma ona 4 parametry i podaje się je jako:

ENT <numer obwiedni>, <ilość kroków>, <ton kroku>,
<czas kroku>

Raz jeszcze prześledźmy znaczenie poszczególnych parametrów.

Numer obwiedni

Jest to parametr, o wartości z zakresu 0-15, wskazujący na numer obwiedni i może być wykorzystany w komendzie SOUND.

Ilość kroków

Parametr <ilość kroków> wskazuje ile różnych tonów ma wystąpić przed zaniknięciem dźwięku. Dla przykładu w nucie, która ma trwać 10 sekund możesz pragnąć dziesięć kroków tonu po 1 sekundzie każdy. W takim wypadku parametr powyższy powinien przyjąć wartość 1Ø.

Parametr może przyjmować wartości od 0 do 239.

Ton kroku

Każdy następny krok może różnić się od poprzedniego o -128 do +127 poziomów dźwięku. Ujemna wartość parametru podwyższa wysokość dźwięku; dodatnia - obniża. Najkrótszy okres tonu wynosi ø, o czym należy pamiętać formułując obwiednię. Cały zakres wysokości dźwięku podany jest w rozdziale "Trochę użytecznych informacji".

Czas kroku

Parametr ten ustala czas trwania kroku w jednostkach setnych części sekundy (0.01 sek). Zakres tego parametru: Ø do 255, co oznacza, że najdłuższy czas trwania jednego kroku wynosi 2,56 sek (Ø traktowane jest jak 256).

Zauważmy zatem, że <ilość kroków> pomnożone przez <czas kroku> nie powinno być dłuższe niż wielkość wynikająca z parametru <czas trwania> komendy SOUND - w przeciwnym razie dźwięk zaniknie przed przejściem przez wszystkie poziomy obwiedni. (W takim wypadku pozostała część obwiedni zostanie pominięta).

Podobnie, jeżeli wielkość wynikająca z parametru <czas trwania> komendy SOUND jest większa niż <ilość kroków> razy <czas kroku> to po przejściu przez wszystkie poziomy obwiedni dźwięk będzie trwał nadal i pozostanie w takim stroju, jaki wynika z ostatniego kroku obwiedni.

Aby lepiej poznać obwiednię tonu wpisz taki program:

```
10 ent 1,100,2,2
20 sound 1,142,200,15,,1
run
```

Linia 20 określa dźwięk o wysokości 142 (międzynarodowe a) trwający 2 sekundy, z początkową głośnością 15 (max), bez obwiedni głośności (co wynika z ,,) i z obwiednią tonu numer 1.

Linia 10 określa obwiednię tonu numer 1, składającą się ze 100 kroków, zwiększającą okres dźwięku (czyli zmniejszającą wysokość) o 2 co 0,02 sek (2 x 0,01).

Pozmieniaj teraz linię 10 na następujące linie:

```
10 ent 1,100,-2,2
10 ent 1,10,4,20
10 ent 1,10,-4,20
```

Zmień teraz komendę SOUND i obwiednię tonu wpisując:

```
10 ent 1,2,17,70
20 sound 1,71,140,15,,1
30 goto 10
run
```

Aby zatrzymać program naciśnij dwa razy ESC .

Teraz już możesz w celu tworzenia różnych dźwięków połączyć obwiednię głośności, tonu i komendę sound.

Zacznij tak:

```
10 env 1,100,1,3
20 ent 1,100,5,3
30 sound 1,142,300,1,1,1
run
```

Następnie zmień linię 20:

```
20 ent 1,100,-2,3
run
```

Po czym zamień wszystkie linie przez wpisanie:

```
10 env 1,100,2,2
20 ent 1,100,-2,2
30 sound 1,142,200,1,1,1
run
```

Jeżeli pragniesz utworzyć ciekawsze obwiednie tonu, to 3 parametry: <ilość kroków>, <ton kroku> i <czas kroku> mogą być na końcu komendy ENT powtórzone do 4 razy, określając różne "sekcje" obwiedni.

Sam spróbuj różnych odmian. Dodaj trochę szumu i kilka dodatkowych sekcji obwiedni tonu i głośności.

Rozdział zatytułowany "Kompletna lista słów kluczowych języka BASIC" zawiera wszystkie detale różnych koment związanych z dźwiękiem.

Jeżeli zainteresowany jesteś melodyczną stroną używania dźwięku, to zajrzyj do części "Niech brzmi muzyka ..." rozdziału "W wolnej chwili...".

Część 10: Wprowadzenie do AMSDOS i CP/M

Czym jest AMSDOS?

Kiedy włączamy komputer do sieci lub wyzerujemy go, to cały system działa pod nadzorem systemu operacyjnego "AMSDOS". AMSDOS jest skrótem od słów AMStrad Disc Operating System (Dyskowy system operacyjny AMSTRAD). Dostępne są w nim następujące komendy manipulujące zbiorami i funkcje:

LOAD "nazwa zbioru"
RUN "nazwa zbioru"
SAVE "nazwa zbioru
CHAIN "nazwa zbioru
MERGE "nazwa zbioru"
CHAIN MERGE "nazwa zbioru"
OPENIN "nazwa zbioru"
OPENOUT "nazwa zbioru"
CLOSEIN
CLOSEOUT
CAT
EOF
INPUT # 9
LINE INPUT #9
LIST #9
PRINT #9
WRITE #9

Dodatkowo AMSDOS zapewnia kilka dodatkowych komend administrujących dyskiem.

Polecenia te zwane są "komendami zewnętrznymi" i oznaczane są przez znak | .

Kilka popularniejszych komend zewnętrznych, które będą z pewnością używane to:

|a
|b
|tape (która to komenda dzieli się na |tape in i |tape.out)
|disc (dzieli się na |disc.in i |disc.out)

Komendy |a i |b używane są w systemach o dwu napędach dyskowych w celu poinstruowania komputera o tym, którego napędu dotyczy następna operacja dyskowa.

Wprowadzenie dla przykładu

|a

load "nazwa zbioru"

mówi komputerowi, aby załadował podany program z dysku w napędzie A.

Jeżeli żadna z komend |a i |b nie została wprowadzona, lub komputer jest wyzerowany - system przyjmuje, że komendy dotyczą napędu A.

Jeżeli używasz jedynie wbudowanego napędu dysków, to należy go traktować jako napęd A, a komendy |a i |b nie powinny być wydawane. Wprowadzenie komendy |b bez podłączonego zewnętrznego napędu doprowadzi do pojawienia się komunikatu o błędzie:

Drive B: disc missing

Retry, Ignore or Cancel

na który należy odpowiedzieć naciskając C (Cancel - skasuj komendę).

A co, jeżeli zechcę użyć kasety? ...

Polecenie |tape informuje komputer, że wszystkie operacje typu ładowania i zapisywania mają działać nie na dysk, lecz na zewnętrzny magnetofon. Tak długo, dopóki nie wprowadzimy komendy |tape, wszystkie operacje na zbiorach odwoływać się będą do dysku.

Aby powrócić do pracy z dyskiem po komendzie |tape należy wprowadzić

|disc

Czasami może zajść potrzeba załadowania programu z kasety i zapamiętaniu go na dysku. Można w takim wypadku użyć polecenia

|tape.in

Nakazuje to komputerowi czytać dane z taśmy, nie zmienia jednak operacji zapisu; będzie on dokonywany na dysk.

Podobnie, aby odczyt dokonywany był z dysku a zapis odbywał się na taśmę, należy wprowadzić |disc.in - aby skasować uprzednią komendę |tape.in - a potem napisać |tape.out po to, aby komputer zapisywał na taśmę.

Można zauważyć, że polecenia |tape.in oraz |tape.out wyłączają działanie poleceń |disc.in i |disc.out - i na odwrót.

Dalsze informacje na temat ukierunkowywania przepływu danych do/z (i kasety) znajdują się w rozdziałach zatytułowanych "Używanie dysków i kaset" oraz "AMSDOS i CP/M"

Kopiowanie programów pomiędzy dyskami (i kasetami)

W części 7 tego Kursu Podstawowego dowiedziałeś się, w jaki sposób sformatować czystą dyskietkę przy pomocy programu DISKIT3 ze strony 1 pakietu dysków systemowych CP/M. Dowiedziałeś się również, jak skopiować program z jednego dysku na drugi. Teraz, przy pomocy poleceń
|tape |disc |tape.in |tape.out |disc.in |disc.out |a |b
będziesz mógł dokonać ładowania i zapisywania zbiorów z dysku umieszczonego albo w napędzie A, albo B (jeżeli ma to zastosowanie) i z kasety (jeżeli ma to sens).

Inne komendy zewnętrzne:
|dir |drive |era |ren |user
opisane są w dalszym ciągu wykładu, w rozdziale "AMSDOS i CP/M"

Kopiowanie całego dysku

Cała zawartość dysku może być skopiowana na inny dysk przy pomocy programu DISCKIT3 ze strony 1 pakietu dysków systemowych CP/M.

W ten sposób można wykonać kopię samych dysków systemowych.

Wprowadź stronę 1 pakietu do napędu w komputerze i napisz
|cpm
Po ukazaniu się A> napisz
disckit3
Po kilku sekundach zobaczysz na środku ekranu nagłówek programu DISC KIT oraz napis
One drive found

Informacje te oznaczają, że działa program DISC KIT oraz że komputer stwierdził, że pracujesz z jednym napędem dyskowym (tym wbudowanym do komputera).

Jeżeli podłączyłeś do komputera dodatkowy napęd to napis będzie następujący:

Two drives found

Na dole ekranu zobaczysz co następuje:

Copy	7
Format	4
Verify	1
Exit from program	Ø

("Kopiuj, Formatu, Weryfikuj, Wyjdź z programu").

Jest to tzw. "menu" główne programu DISC KIT. Numerki w ramkach odpowiadają numerom klawiszy funkcyjnych na bocznej klawiaturze (po prawej stronie) - oznaczonym fØ, f1, f4, i f7. Naciskając jeden z tych klawiszy dokonujesz wyboru funkcji wykonywanej przez program. Zwróć uwagę, że naciśnięcie klawisza o numerze Ø na tym etapie działania programu spowoduje powrót do trybu oczekiwania na komendę systemu CP/M (znaki A>). Ponieważ pragniemy skopiować dysk - naciśnijmy klawisz funkcyjny o numerze 7 (f7).

OSTRZEŻENIE

Kopiowanie na uprzednio zapisany dysk wymaże jego starą zawartość

Kopiowanie w systemie z 1 napędem

Przyjmując, że używasz jednego napędu (tzn. NIE podłączyłeś dodatkowej stacji dysków), napis który zobaczysz będzie następujący:

Y Copy

Any other key to exit menu

("Kopiuj","Jakikolwiek inny klawisz w celu wyjścia do menu"). W tej chwili należy wyjąć dysk systemowy CP/M i włożyć dysk, który pragniesz skopiować. Jeżeli chcesz skopiować sam dysk systemowy, to pozostaw go w napędzie.

Gdy dysk, z którego odbywać się będzie kopiowanie jest już w napędzie, naciśnij Y (Y - jako skrót od Yes - tak) - co oznacza "jedź dalej i kopiuj dysk".

Format w jakim zapisany jest dysk zostanie na wstępie zbadany, a odpowiednia informacja pojawi się na górze ekranu.

Po chwili zobaczysz napis:

Insert disc to WRITE
Press any key to cintinue

("Wprowadź dysk do zapisu, naciśnij dowolny klawisz w celu kontynuowania"). W tym momencie należy wyjąć dysk z napędu i zastąpić go dyskiem na który chcemy zapisać dane; następnie nacisnąć jakikolwiek klawisz.

Na gorze ekranu pojawi się informacja o formacie, w jakim zapisany zostanie nowy dysk nawet, jeżeli kopiujemy na nową dyskietkę.

Jeżeli dysk na który odbywa się kopiowanie nie jest sformatowany lub sformatowany jest błędnie, to na ekranie pojawi się napis (taki, lub podobny w zależności od dysku źródłowego):

Disc isn't formatted (or faulty)
Going to format while copying
Disc will be system format

co oznacza, że formatowanie będzie miało miejsce podczas kopiowania.

Kiedy już komputer gotów jest do odczytania porcji informacji z dysku źródłowego, to wyświetlony zostanie napis:

Insert disc to READ
Press any key to continue

i wtedy należy z powrotem umieścić w napędzie dysk z którego kopiujemy.

Po kilkukrotnym powtórzeniu tych operacji cała zawartość dysku źródłowego zostanie przepisana na nowy dysk a napis

Copy completed
Remove disc
Press any key to continue

poinformuje, że kopiowanie dysku jest zakończone, należy opróżnić napęd i nacisnąć dowolny klawisz. Można następnie skopiować następną stronę dysku (naciskając klawisz Y) lub przejść do głównego "menu" programu DISC KIT.

Zabezpieczenie przed zapisem

Należy zwrócić uwagę, że nie można kopiować na dysk, którego otwór zabezpieczający przed zapisem jest otwarty. Każda próba kopiowania na taki dysk da rezultat w postaci napisu

Disc write-protected
Insert disc to write
R-etry or C-ancel

po ukazaniu się którego należy nacisnąć klawisz C, wyjąć dysk i włożyć nowy dysk z zamkniętym otworem zabezpieczającym. Dopilnuj, aby otwory na dyskach zawierających ważne programy nigdy nie były zamknięte. W szczególności NIGDY nie należy zasłaniać tych otworów na wzorcowych dyskach systemowych CP/M.

Kopiowanie w systemie z dwoma napędami

Należy kierować się powyższymi wskazówkami i załadować program DISC KIT3 ze strony 1, a następnie wybrać funkcję COPY poprzez naciśnięcie klawisza f7

W tym miejscu na ekranie pojawi się dalsze "menu"

Read from A:	8
Read from B:	5
Exit menu	2

("Czytaj z A:", "Czytaj z B:","wyjdź z menu").
Kopiowanie w systemie z dwoma napędami oznacza, że nie trzeba ciągle powtarzać operacji wymiany dysków. Powyższe menu pozwala wybrać napęd do którego włożony będzie dysk źródłowy. Naciśnij f8 w celu kopiowania z dysku w napędzie A:
Zostanie wtedy wyświetlone trzecie menu:

Write to A:	9
Write to B:	6
Exit menu	3

("Pisz na A:", "Pisz na B:", "wyjdź z menu"). Menu to pozwala na wybranie napędu, w którym znajdować się będzie dysk przeznaczony do zapisu. Możesz, jeżeli tylko chcesz, wybrać napęd A:,

ale oznaczać to będzie potrzebę ciągłego wymieniania dysków i nie odniesiesz żadnej korzyści z posiadania systemu z 2 napędami.

Dla wygody i z uwagi na szybkość kopiowania należy wybrać napęd B:, naciskając klawisz f6.

Teraz należy umieścić oba dyski we właściwych napędach i nacisnąć klawisz Y - sygnał do rozpoczęcia kopiowania.

I znów na ekranie pojawi się informacja o formacie obydwu dysków, a jeżeli dysk docelowy jest sformatowany błędnie (lub w ogóle nie jest) to dokona się formatowanie podczas kopiowania.

Po zakończonym kopiowaniu na ekranie pojawi się następujący napis

Copy completed
Remove both discs
Press any key to continue

po czym należy wyjąć OBA dyski, zanim podejmie się jakiekolwiek inne działanie. Następnie można powtórzyć operację kopiowania przez wciśnięcie klawisza Y, lub przejść do głównego menu programu DISC KIT naciskając jakikolwiek inny klawisz.

Weryfikacja dysków

Program DISCKIT3 pozwala również na weryfikowanie (sprawdzanie) dysku.

Na ekranie wyświetlana jest informacja o formacie dysku, czytane są wszystkie zbiory na dysku i w wypadku znalezienia błędu odpowiednia informacja pojawia się na ekranie.

Aby zweryfikować dysk należy do napędu włożyć dysk systemowy oznaczony cyfrą 1 i napisać

|cpm

Po pojawieniu się znaków A>, napisz

disckit3

Wybierz opcję Verify z głównego menu (klawisz f1) a następnie kieruj się wskazówkami na ekranie.

Jeżeli pracujesz z systemem o 2 napędach, to będziesz mógł wybrać, czy weryfikowaniu podlegać będzie dysk w napędzie A czy B.

Kied[illegible] już we wła[illegible]wym napędzie znajdzie się odpowiedni dysk [illegible] naciśnij Y aby zapoczątkować proces weryfikacji.

[illegible]s[illegible]czonym sprawdzaniu na ekranie ukaże się:

```
r[illegible]f [illegible]omple[illegible]d
mo[illegible]e [illegible]
P[illegible]es[illegible] any k[illegible] [illegible]o co[illegible]tinue
```

Jeżeli pracujesz w systemie z 2 napędami to należy wyjąć ob[illegible] dyski. Trzeba to zrobić przed jakąkolwiek inną czynnością. [illegible]nas[illegible]pnie powtórzyć proces naciskając Y lub przejść do głównego menu naciskając dowolny, inny klawisz.

Używanie programu DISC KIT pod CP/M 2.2

Jeżeli zechcesz sformatować lub zweryfikować dyski zapisane na lub dla CPC 664 lub CPC 464+DDI1 (oba te zestawy używają CP/M 2.2), na stronie 4 pakietu dysków systemowych dołączono program DISC KIT. Nazywa się on DISCKIT2 i działa w ten sam sposób jak powyżej opisany DISCKIT3.

Aby uruchomić DISCKIT2 należy użyć strony 4 pakietu dysków systemowych i napisać

```
|cpm
```

Po ukazaniu się "nagłówka" CP/M 2.2 i znaków gotowości A> napisz

```
disckit2
```

Od tej chwili po prostu wybierz odpowiednią opcję z "menu" i postępuj zgodnie ze wskazówkami na ekranie tak, jak w programie DISCKIT3.

UWAGA: Program DISCKIT2 podczas kopiowania używa obszaru pamięci ekranu do przechowywania danych. Podczas kopiowania zobaczysz zatem na ekranie losowe wzory.

Dalsze informacje

Dalsze informacje o sposobie używania programów znajdujących się na wzorcowych dyskach CP/M będą podane w dalszej części podręcznika .

Na koniec upewnij się, że zastosowałeś się do ostrzeżeń podanych na początku podręcznika,w sekcjizatytułowanej WAZNE

Uwagi o instalacji   5,6,7

Uwagi o obsłudze   1,2,3,4,5,6,7,9

Część 11: Wprowadzamy program "Bank Menager" ...

Używanie drugich 64 KB pamięci

CPC zawiera 128 KB pamięci RAM (Random Access Memory - pamięć o swobodnym dostępie, do zapisu i odczytu informacji) w dwóch częściach po 64 KB. CP/M Plus używa cały czas pełnych 128 KB, lecz BASIC normalnie nie używa drugich 64 KB - dysponuje tylko pamięcią w pierwszych 64 KB. Byłoby żal zostawić dodatkowe 64 KB pamięci zupełnie nieużywane przy programowaniu w języku BASIC, a zatem dostarczono program umożliwiający w szczególnych przypadkach korzystanie z tej dodatkowej pamięci. Program ten oferuje kilka dodatkowych rozkazów za pomocą których można wykorzystać drugie 64 KB RAM do przechowywania obrazów ekranu lub jako przestrzeń do przechowywania ciągów tekstowych.

Program wprowadzający takie dodatkowe rozkazy został nazwany "BANK MENAGER", gdyż "bank" jest terminem technicznym stosowanym do określenia zespołu pamięci.

Zastosowanie programu "BANK MENAGER" do przechowywania obrazów ekranu

6128 wyświetla obraz na ekranie przez cały czas. Do tego celu używa 16 KB pamięci w której przechowywane są informacje o kolorze i jasności każdego pojedynczego punktu ekranu. Pamięć 6128 ma możliwość jednoczesnego przechowywania do 6 obrazów ekranu (każdy w osobnym bloku 16 KB). BANK MENAGER wprowadza udogodnienia pozwalające w języku BASIC na swobodne żonglowanie pięcioma z sześciu możliwych obrazów ekranu.

Bezpośrednio po włączeniu komputera obraz na ekranie jest wyświetlany z 16 KB bloku pamięci (który będziemy nazywać "Blokiem 1"), wydzielonego z pierwszych 64 KB. Pozostałe cztery obrazy ekranu są przechowywane w drugich 64 KB pamięci, w blokach nazywanych Blok 2, Blok 3, Blok 4 i Blok 5.

Tylko Blok 1 (z pierwszych 64 KB) może być w y ś w i e t l a n y na ekranie. Aby z o b a c z y ć obrazy przechowywane w drugich 64 KB (Bloki 2 do 5) należy przenieść żądany obraz do Bloku 1. BANK MENAGER wprowadza wszystkie rozkazy

potrzebne do przenoszenia obrazów jak SCREENCOPY, który po prostu przenosi treść obrazu do innego bloku i SCREENSWAP, który wymienia zawartość dwóch obrazów.

Podobnie jak rozkazy AMSDOS, omówione wcześniej w tym rozdziale, BANK MENAGER używa "rozkazów zewnętrznych", które rozpoczynają się od symbolu kreski | (uzyskiwanego przez naciśnięcie [SHIFT]@).

Jak używać BANK MENAGER

Wyzeruj komputer za pomocą [CONTROL][SHIFT][ESC] a następnie wprowadź Stronę 1 ze swojego pakietu dysków systemowych i napisz:

```
RUN "BANKMAN"
```

Procedura ładowania jest szczegółowo opisana w części 13 Rozdziału 7, dotyczącej rozszerzeń systemu rezydendnego (RSX) i byłoby korzystnie rozumieć nieco technikę RSX i sposoby rezerwowania obszarów pamięci przed użyciem tych procedur w Twoich programach. Jednakże do wypróbowania poniższych przykładów nie musisz wiele wiedzieć o procedurze ładowania.

Napisz:

```
MODE 1
PRINT "TO JEST ORYGINALNY EKRAN"
|SCREENSWAP,1,2
```

Tekst zniknie, a zamiast niego będziesz oglądał to, co schowano wcześniej jako Ekran 2 (w Bloku 2). Jeśli komputer został przed chwilą włączony, będzie to prawdopodobnie abstrakcyjny, losowy wytworzony wzór. Aby usunąć ten wzór, wprowadź:

```
MODE 1
```

a następnie:

```
PRINT "TO JEST EKRAN 2"
|SCREENSWAP,1,2
```

Pojawia się znowu Twój oryginalny tekst. Jeśli teraz powtórzysz polecenie |SCREENSWAP,1,2 możesz zobaczyć, że wymieniona będzie zawartość dwóch ekranów. Możesz wymienić (SWAP) zawartość dowolnych z pięciu ekranów, jednakże miej na uwadze, że tylko gdy dokonujesz wymiany obejmującej Ekran 1, zmienia się treść w y ś w i e t l a n a na ekranie.

Innym dostępnym teraz rozkazem jest |SCRFENCOPY. Powoduje on kopiowanie jednego ekranu w miejsce innego, zastępując tam starą treść nowym obrazem.
Wprowadź:

```
MODF 1
PRINT "TO JEST EKRAN DO KOPIOWANIA
|SCRFENCOPY,2,1
```

Zawartość Ekranu 1 jest kopiowana do Ekranu 2. Jeżeli odwrócisz parametry i napiszesz:

```
MODF 1
|SCRFENCOPY,1,2
```

treść aktualnie wyświetlana na ekranie zostanie zastąpiona zawartością Ekranu 2.

Pierwszy parametr określa ekran, do którego następuje kopiowanie, drugi parametr oznacza ekran z którego kopiuje się treść.

Przy kopiowaniu ekranów można otrzymać poszarpany, bezładny obraz jeżeli różne są parametry MODE ekranów lub gdy wyświetlany obraz został "zrolowany" po ostatnim poleceniu MODE. Instrukcje ekranowe programu BANK MENAGER są przeznaczone do stosowania głównie do ekranów graficznych (obrazkowych) a nie tekstowych, a w przypadku ekranów graficznych rolowanie ekranu nie jest stosowane.

Używanie programu BANK MENAGER do przechowywania ciągów tekstowych

BANK MENAGER wprowadza dalsze cztery rozkazy, które umożliwiają użycie dodatkowych 64 KB pamięci do przechowywania zmiennych tekstowych.

Większość programów może być podzielona na dwie części: pierwsza zawiera aktualne instrukcje programu a druga to dane, używane przez program. Dobrym przykładem takiego podziału jest program obsługi bazy danych jak np. książka adresowa. Program taki mógłby używać tablic ciągów tekstowych do przechowywania nazwisk i adresów ludzi umieszczonych w książce.

Ciągi tekstowe mogą być przechowywane w drugich 64 KB pamięci, bezpośrednio jeden za drugim. Pamięć, w której ciągi

są przechowywane może być dzielona na przedziały, nazywane rekordami . Rekord musi mieć określoną stałą długość od 2 do 255 znaków, chociaż długość ciągu w języku BASIC zmienia się zależnie od jego zawartości. Celem stosowania rekordów jest wprowadzenie przedziałów o standardowych rozmiarach, podobnych do zespołu klatek dla gołębi, w których przechowywana jest jednostkowa część zawartości ciągu. Po każdej operacji zapisu lub pobrania danych z rekordu automatycznie jest przeprowadzane przesunięcie do następnego rekordu, gotowego do następnej operacji. Rekord przygotowany w ten sposób do kolejnej operacji nazywany jest "rekordem bieżącym" i jest używany automatycznie jeżeli nie zostanie wyspecyfikowany inny rekord.

Taki sposób gospodarowania pamięcią tworzy strukturę nazywaną "dyskiem typu RAM", gdyż struktura taka działa w podobny sposób jak system dyskowy o swobodnym dostępie, używając tylko pamięci RAM zamiast dysku.

Przeczytaj poniższe opisy różnych rozkazów,tak, abyś rozumiał, do czego poszczególne rozkazy służą, nawet gdybyś nie wiedział jak je użyć, a następnie wypróbuj podane przykłady.

Pierwszym rozkazem operującym z "dyskiem typu RAM" jest |BANKOPEN. Określa on ile znaków ciągu tekstowego zawierać może każdy rekord. Składnia tego rozkazu jest następująca:

|BANKOPEN,n

gdzie n jest liczbą określającą liczbę znaków w rekordzie. Wartość n może leżeć w zakresie od 0 do 255, lecz wartości 0 i 1 mogą powodować dziwne efekty.

|BANKWRITE zapisuje ciąg do bieżącego rekordu. Wskaźnik bieżącego rekordu jest następnie zwiększany o 1 tak aby wskazywał następny rekord za rekordem właśnie zapisanym, gotów do następnej operacji. Składnia tego rozkazu jest następująca:

|BANKWRITE, r%, a$

lub

|BANKWRITE,@r%,a$,n

gdzie r% jest zmienną typu liczby całkowitej do której wpisywany jest kod informujący o poprawności wykonania operacji a a$ jest zmienną tekstową, zawierającą znaki wpisywane do rekordu. W pierwszym z dwu przykładów rekord jest rekordem

bieżącym. W drugim przykładzie opcjonalny parametr n określa rekord, do którego następuje wpisywanie.

|BANKREAD czyta bieżący rekord i wstawia jego zawartość do określonej zmiennej tekstowej. Wskaźnik rekordu bieżącego jest następnie zwiększany o 1 tak aby wskazywał następny rekord, bezpośrednio za rekordem właśnie odczytanym, gotowy do następnej operacji. Odczytanie rekordu nie zmienia jego zawartości tak, że rekord może być czytany powtórnie wiele razy. Składnia rozkazu |BANKREAD jest następująca:

BANKREAD,∂r%,a$

lub:

|BANKREAD,∂r%,a$,n

gdzie r% jest zmienną typu liczby całkowitej do której wpisywany jest kod informujący o poprawności wykonania operacji a a$ jest zmienną tekstową do której wpisywana jest odczytywana zawartość rekordu. W pierwszym z dwu przykładów rekord jest rekordem bieżącym. W drugim przykładzie opcjonalny parametr n określa rekord który ma być odczytany.

Ostatnim rozkazem tego typu jest |BANKFIND. Rozkaz ten powoduje przeszukiwanie rekordów w celu znalezienia określonego ciągu tekstowego. Jeżeli ciąg zostanie znaleziony, podawany jest numer rekordu w którym znajduje się zadany ciąg. Składnia tego rozkazu jest następująca:

|BANKFIND,∂r%,a$

lub:

|BANKFIND,∂r%,a$,n

lub

|BANKFIND,∂r%,a$,n,m

gdzie r% jest zmienną typu liczby całkowitej do której wpisywany jest numer rekordu w którym znaleziono zadany ciąg lub kod wskazujący, że ciągu nie znaleziono a a$ jest szukanym ciągiem. Opcjonalny parametr n podaje numer rekordu od którego zaczyna się poszukiwanie. Jeżeli n jest pominięte, przeszukiwanie rozpoczyna się od bieżącego rekordu. Drugi parametr opcjonalny m określa numer rekordu w którym kończy się poszukiwanie nawet jeżeli nie zostanie jeszcze znaleziony odpowiednik! Jeżeli m jest pominięte, poszukiwanie kontynuowane jest przez całe 64 KB, co może wykraczać poza koniec wpisanych danych.

A teraz wypróbuj to w działaniu. Jeżeli masz już wprowadzony program BANK MENAGER, gdyż przyglądałeś się działaniu rozkazów wymiany ekranu i od tego czasu nie zerowałeś komputera to omawiane wyżej rozkazy rezydują już w pamięci komputera. Jeżeli nie, to wprowadź Stronę 1 ze swojego pakietu dysków systemowych i napisz:

```
RUN "BANKMAN"
```

Następnie napisz:

```
|BANKOPEN,2Ø
```

Określa to długość rekordu o wartości 20 znaków i ustawia bieżący numer rekordu równy 0.

Teraz napisz:

```
a$ = "PIERWSZY ZAPIS" + SPACE$(6)
```

co wpisuje do zmiennej aS dokładnie 20 znaków.

Następnie wprowadź:

```
r% = Ø
```

aby zainicjalizować zmienną r%.

Z kolei napisz:

```
|BANKWRITE,∂r%,a$
```

Powinno to zapisać aS do pierwszego rekordu (rekord Ø).

Teraz napisz:

```
d$=SPACE$(2Ø)
|BANKREAD,∂r%,d$,Ø
PRINT d$
```

Pierwszy rozkaz wprowadza zmienną d$, zawierającą 20 spacji. Stwarza to wystarczające miejsce aby pomieścić zawartość rekordu kiedy zostanie odczytany. Drugi rozkaz czyta rekord numer 0 i wprowadza rezultat do d$. Ponieważ bieżący numer rekordu został przesunięty do rekordu 1 (przez poprzednią operację |BANKWRITE), konieczne jest określenie rekordu 0 jako rekordu do czytania. Pamiętaj - jeżeli numer rekordu nie jest wyspecyfikowany, czytany jest bieżący rekord. Na koniec wyświetlany jest rezultat operacji czytania. Zatem zmienna d$ powinna zawierać PIERWSZY ZAPIS i 6 spacji.

A teraz napisz:

```
b$ = "DWA + SPACE$(17)
c$ = "TRZY + SPACE$(16)
|BANKWRITE,∂r%,b$,1
|BANKWRITE,∂r%,cS
```

Wpisuje to b$ i c$ do rekordów 1 i 2. Znowu użyto parametru opcjonalnego w pierwszym BANKWRITE, aby określić rekord 1 jako rekord, do którego wpisuje się b$. Bieżący numer rekordu jest następnie przesuwany do następnego rekordu tak, że niepotrzebne było specyfikowanie do którego rekordu należy wpisać c$ - zmienna ta zostanie automatycznie wpisana do rekordu 2.

Napisz:

```
PRINT r%
```

Wyświetla to wartość r%, która po wykonanym przykładzie powinna być równa 2. Liczba ta może być interpretowana jako numer rekordu z którym prowadzono ostatnią operację lub jako numer rekordu bieżącego minus jeden. W powyższym przykładzie ostatnia operacja prowadzona była z rekordem 2 a następnym rekordem jest 3.

Celem takiego "kodu powrotnego" jest dostarczanie informacji o właśnie przeprowadzonej operacji. Po pomyślnie zakończonej operacji "kod powrotny" powinien mieć wartość dodatnią, wskazującą numer rekordu; operacja zakończona nieprawidłowo powoduje wpisanie liczby ujemnej reprezentującej kod błędu. Po operacjach BANKWRITE i BANKREAD otrzymać można dwie wartości kodu błędu:

- -1 Wskazuje na osiągnięcie końca zbioru. Zdarza się gdy zużyte zostały wszystkie rekordy lub wyspecyfikowano rekord który nie istnieje
- -2 Sygnalizuje błąd przełączania banku. Błąd taki nie powinien nigdy wystąpić.

Wypróbuj teraz dalsze przykłady:

```
d$ = STRING$(2Ø,"X")
|BANKOPEN,2Ø
FOR n=1 to 3: BANKWRITE, r%,d$:NEXT
```

Wprowadza to zmienną d$, zawierającą 20 znaków X. |BANKOPEN zeruje wskaźnik rekordu bieżącego a rozkaz |BANKWRITE wpisuje d$ do rekordów 0, 1 i 2 w miejsce poprzedniej zawartości tych rekordów.

Teraz napisz:

```
a$ = "PIERWSZY"
|BANKWRITE, r%,a$,Ø
```

Wpisuje to słowo "PIERWSZY" do rekordu 0 w miejsce niektórych znaków X. Wypełń teraz zmienną d$ spacjami:

```
d$ = SPACE$(2Ø)
```

i napisz:

```
|BANKREAD,∂r%,d$,Ø
```

Wczytałeś rekord Ø z powrotem do d$.

Przypomnijmy sobie tok postępowania:

Wszystkie trzy rekordy wypełnione zostały X-ami. Do rekordu 0 dodano słowo "PIERWSZY"; d$ wypełniono spacjami i na koniec odczytano rekord 0 do d$. Napisz zatem:

```
PRINT d$
```

W wyniku powinieneś otrzymać "PIERWSZYXXXXXXXXXXXX"

Ilustruje to ważną właściwość omawianych tutaj rozkazów. Jeżeli ciąg umieszczany w rekordzie nie zapełnia rekordu całkowicie, pozostają w nim stare znaki, które nie zostały zastąpione nowymi. Jest zatem wskazane aby przed zapisaniem świeżej informacji do rekordu wypełnić go najpierw spacjami albo znakami CHR$(32). Unika się w ten sposób stwierdzenia, że w odczytanym z powrotem ciągu znajdują się znaki, które nie powinny w nim już występować. Te same rozważania dotyczą zmiennej tekstowej do której rekord jest wczytywany. Jeżeli zmienna tekstowa jest dłuższa od długości rekordu to na końcu tej zmiennej pozostaną niezmienione znaki. Z tego powodu czyściliśmy (wypełnialiśmy spacjami) zmienną d$ przed odczytaniem do niej rekordu 0.

Jest możliwe wpisanie ciągu tekstowego do rekordu, który jest zbyt krótki aby pomieścić cały ciąg. Jeżeli to nastąpi, nadmiarowe znaki zostaną zignorowane (pominięte).

Jest również możliwe odczytanie długiego rekordu do krótkiej zmiennej tekstowej i także w tym przypadku zostaną zignorowane nadmiarowe znaki (odczytywane z rekordu). Przy normalnych operacjach ze zmiennymi tekstowymi BASIC mógłby automatycznie rozciągnąć zmienną tekstową tak aby zmieścić nadmiarowe znaki, nie może to jednak nastąpić przy wykonywaniu rozkazów zewnętrznych.

Na koniec rozkaz |BANKFIND. Umożliwia on przeszukiwanie rekordów w celu znalezienia określonego etekstu. Na przykład, jeśli rekord numer 24 zaczyna się od słowa "FRED", może to być znalezione za pomocą polecenia:

```
|BANKFIND,∂r%,"FRED"
```

Jest to szczególnie użyteczne w programach obsługi bazy danych w których np. wyszukiwane są nazwiska lub adresy.

|BANKFIND zaczyna od bieżącego rekordu i prowadzi swoje działania poprzez kolejne rekordy aż do znalezienia szukanego ciągu tekstowego lub do końca drugich 64 KB pamięci, w których rekordy są przechowywane.

Można określić numer rekordu od którego rozpoczyna się poszukiwania przez dołączenie numeru rekordu na końcu rozkazu. Można ponadto dołączyć jeszcze jeden numer rekordu, umieszczony po numerze rekordu początkowego, aby określić rekord, na którym kończą się poszukiwania.

|BANKFIND może być użyty do wyszukiwania ciągu tekstowego, który nie leży na początku rekordu. W takim przypadku należy umieścić znaki CHR$(Ø) przed znaczącymi znakami poszukiwanego ciągu. Znaki CHR$(Ø) są traktowane jako znaki zastępcze (jak ? w nazwach zbiorów w systemie CP/M) co oznacza, że znaki te są liczone jako "jakikolwiek znak". Jako przykład powyższego może służyć:

```
a$ = STRING$(1Ø,Ø)+"FRED"
|BANKFIND,@r%,a$,Ø
```

Powinno się w ten sposób znaleźć słowo "FRED" umieszczone od pozycji jedenastego do pozycji czternastego znaku rekordu. Pierwsze dziesięć znaków może zawierać numer telefonu lub jakieś inne informacje, które będą ignorowane przez BANKFIND.

W przypadku znalezienia słowa "FRED" pod zmienną r% podstawiany jest numer rekordu w który znaleziono zadany tekst, w przeciwnym przypadku podstawiany jest kod powrotny o wartości -3.

Dalsze szczegóły ...

Dalsze informacje o programie BANK MENAGER można znaleźć w Rozdziale 8; mógłbyś także przestudiować część 13 i 14 Rozdziału 7, gdzie opisano zagadnienia związane z zastosowaniem RSX-ów.

Na koniec upewnij się czy zastosowałeś się do następujących ostrzeżeń podanych na początku tego podręcznika w części zatytułowanej "WAŻNE":

Uwagi o obsłudze 1,2,3,9

Kończy to II część "Kursu podstawowego" posługiwania się CPC 6128. Teraz powinieneś już wiedzieć jakie jest działanie większości klawiszy klawiatury, jak używać pewne najprostsze rozkazy BASIC, jak formatować fabrycznie nowy dysk tak aby był gotów do użycia i jak wykonać pewne najbardziej elementarne operacje dyskowe: wprowadzanie (LOAD) i zapisywanie (SAVE) zbiorów, odczytywanie skorowidzu (CAT) i kilka rozkazów AMSDOS, CP/M i BANK MENAGER.

W następnych rozdziałach znajdziesz bardziej specjalizowany opis niektórych zagadnień techniki komputerowej i języka AMSTRAD BASIC. Jest tam także szerszy opis posługiwania się dyskami i systemów AMSDOS oraz CP/M a ponadto możesz wejść w dziedzinę nowego języka - Dr. LOGO firmy Digital Research.

Powodzenia i przyjemnego czytania!

ROZDZIAŁ 2
PO KURSIE PODSTAWOWYM...

Przeczytałeś już Rozdział 1 i masz przed sobą włączony komputer. Nauczyłeś się już jak kazać mu powtarzać zadaną operację kilka razy za pomocą pętli FOR NEXT i jak sprawdzać IF (czy) warunek jest spełniony, a jeżeli - THEN (wtedy) wykonywać określoną czynność.

Jesteś już chyba zmęczony oglądaniem swojego imienia wyświetlanego na całym ekranie i chciałbyś rozpocząć poważną pracę z komputerem - zrobić coś użytecznego lub zabawnego. W rozdziale następnym jest do Twojej dyspozycji wykaz wszystkich słów kluczowych języka AMSTRAD BASIC łącznie z opisem składni (syntaktyki) i wyjaśnieniem do czego ich używać. Po uzbrojeniu w tę listę, zakres tego, co możesz kazać komputerowi wykonać jest ograniczony tylko przez Twoją wyobraźnię.

Jeżeli nie używałeś nigdy przedtem komputera, myśl o rozpoczęciu "programowania" może napełnić Cię obawami. Nie przejmuj się! To jest znacznie łatwiejsze niż myślisz, a na pewno znacznie łatwiejsze niż technologia i żargon, któremu musiałbyś zawierzyć. Myśl o języku BASIC nie jak o nowym sztucznym języku, lecz jak o odmianie języka angielskiego z niektórymi słowami skróconymi dla przyspieszenia komunikowania. Innymi słowy nie myśl np. o CLS jak o magicznym kodzie 3-literowym, lecz pomyśl sobie CLear Screen - wyczyść ekran.

Próbuj nie bać się języka BASIC a szybko znajdziesz zadowolenie zarówno w sztuce programowania jak i wynikach swoich usiłowań. Programowanie może być bardzo wdzięcznym ćwiczeniem, zwłaszcza jeżeli jesteś początkującym w działaniach z komputerem i językiem. Stale pamiętaj, że jeżeli zapewnisz, że przypadkowo nic nie zapiszesz na swoich głównych dyskach systemowych, nic co piszesz nie może uszkodzić komputera i w najgorszym przypadku będziesz musiał spróbować coś nowego.

A zatem od czego zacząć?

Rozpoczęcie jest często najtrudniejszą częścią programowania dla początkującego. Jednakże to, czego powinieneś naprawdę unikać, to bezmyślnego kiwania się nad klawiaturą.

Jedną z pierwszych rzeczy, jaką powinieneś ustalić jest co dokładnie chcesz aby program wykonywał i jak chcesz aby były przedstawiane Ci wyniki (innymi słowy, jak powinien wyglądać ekran po "zapuszczeniu" programu).

Jeżeli zdecydowałeś już to, możesz rozpocząć pisanie programu spełniającego Twoje wymagania, cały czas myśląc jak zapewnić aby program przebiegał gładko od początku do końca z minimalną liczbą odskoków, używających GOTO tu i GOTO tam. Dobry program powinien być łatwy do nadążania za nim przy listowaniu i nie powinien Cię doprowadzać do beznadziejnego zamętu w głowie, kiedy próbujesz znaleźć w nim błąd, czyli "de-bug'ować" (odpluskwiać) program, jak mówi się w żargonie komputerowym.

Na szczęście, BASIC jest łatwo przebaczającym językiem i często będzie Ci pomagał, wyświetlając na ekranie komunikaty o błędzie, jeżeli postąpisz źle. BASIC także umożliwia Ci "reflektować się" i upychać nowe linie programu pomiędzy istniejące linie z minimalnym towarzyszącym temu zamieszaniem.

Piszmy prosty program

A zatem do roboty. Będziemy pisać program do przechowywania wykazu nazwisk Twoich przyjaciół i ich numerów telefonów. Nazwiemy ten program "Książką telefoniczną". A teraz zastosujmy przytoczoną wyżej regułę: "Co program powinien robić?" i "Jak powinny być przedstawiane rezultaty?"

Załóżmy, że program powinien Ci umożliwić wprowadzenie, powiedzmy, do 100 nazwisk i numerów telefonicznych do przechowania. Jeżeli chcesz jakiś numer, powinieneś móc napisać nazwisko i otrzymać numer z powrotem. Dodatkowo, jeżeli nie jesteś pewien jak jakieś nazwisko zostało wprowadzone, powinieneś móc wyświetlić kompletną listę wszystkich informacji na ekranie. Zauważ przy okazji, że automatycznie zaczęliśmy rozważać kwestię, jak prezentowane będą wyniki działania programu.

A więc połóż palce na klawiaturze! Zaczniemy od tytułu na początku:

```
10 REM książka telefoniczna
```

Nie musiałeś umieszczać tytułu w programie, ale kiedy zaczniesz gromadzić co najmniej kilka programów, pomoże Ci to poznać na pierwszy rzut oka który jest który.

Następnie wiemy, że chcemy móc wprowadzić - INPUT - ciąg znaków (czyjeś nazwisko) do zmiennej; nazwiemy tę zmienną NAME$. To samo tyczy numeru telefonu, nazwiemy zatem drugą zmienną TEL$.

Pamiętasz przykładowy program z "Kursu podstawowego"? Używano w nim rozkazu INPUT do wprowadzenia wartości do zmiennej, a zatem możemy napisać:

```
20 INPUT "w p r o w a d ź   n a z w i s k o"; NAME$
30 INPUT "w p r o w a d ź   n u m e r   t e l e f o n u"; TEL$
r u n
```

Możesz teraz wprowadzić nazwisko, na przykład: Józek. Możesz następnie wprowadzić numer telefonu, na przykład: 0277 230222.

Program zapamiętał informacje, lecz nie może wyświetlić na ekranie rezultatu. Potrzebna jest teraz część programu, umożliwiająca odzyskanie informacji, a zatem jej wyświetlenie. Do otrzymania wartości NAME$ i TEL$ w tym momencie, możemy użyć następujących rozkazów:

PRINT NAME$ ... i ... PRINT TEL$

Ale co tam! Powiedzieliśmy że chcemy móc przechować aż do 100 nazwisk i 100 numerów telefonów. Na pewno nie musimy pisać programu z dwustoma rozkazami INPUT, każdy z odmienną nazwą zmiennej i dwustoma rozkazami PRINT do wyświetlenia listy na ekranie ??? Nie martw się, komputer ma do tego specjalne urządzenia, które nazywają się "tablicami" (array). Tablica umożliwia Ci użycie jednej zmiennej (takiej jak np. NAME$), która może mieć dowolną liczbę wymiarów (w naszym programie wymagamy 100). Jeżeli przy tym chcesz otrzymać wartość zmiennej, musisz użyć nazwy zmiennej z umieszczonym za nią numerem porządkowym (w nawiasach). Taki numer porządkowy nazywany jest indeksem a wyrażenie jak np. NAME$ (27) nazywa się zmienna indeksowana. Jeżeli teraz użyjemy zmiennej liczbowej jako indeksu, dla przykładu NAME$ (x), możemy następnie obsłużyć całą listę zmiennych od 1 do 100 przez zmianę wartości x w pętli FOR NEXT,

tj. FOR x=1 TO 100. Ponieważ wartość x przyrasta o 1, to wartość indeksu zmienia się i przemiata wszystkie elementy czy nazwiska w tablicy NAME$.

My potrzebujemy dwóch tablic: jednej dla NAME$ i drugiej dla TEL$, każdej o wymiarze 100 elementów. Przed rozpoczęciem używania tablicy, musimy na początku zadeklarować jej wymiary za pomocą instrukcji DIM (DIMensions=wymiary). Napiszemy takie instrukcje w miejsce dotychczasowych w liniach 20 i 30:

```
20 DIM NAME$(100)
30 DIM TEL$(100)
```

Mając ustalone nasze zmienne, napiszmy kawałek programu który na początek umożliwi nam wprowadzenie nazwisk i numerów telefonów do tablic w celu ich późniejszego odzyskania. Dodaj:

```
40 FOR x=1 TO 100
50 INPUT;"n a z w i s k o"; NAME$(x)
60 INPUT " t e l e f o n "; TEL$(x)
70 NEXT
run
```

To jest wszystko bardzo dobrze, ale my możemy nie chcieć wprowadzać wszystkich 100 nazwisk jednocześnie. Co więcej, fakt że program prezentuje sam siebie na ekranie jest bardziej niż niezadowalający. Aby usunąć te niedogodności, wystarczy drobna kosmetyka. Po pierwsze, przed rozpoczęciem wprowadzania każdego nowego nazwiska i numeru, oczyśćmy ekran z całego poprzedniego zbytecznego już tekstu przez użycie za każdym razem rozkazu CLS (CLear Screen). Dodaj:

```
45 CLS
```

Teraz jak powiadomić komputer, że właśnie zakończyliśmy wprowadzanie informacji? Naciśnięcie [ESC] może zatrzymać program, ale skoro tylko wprowadzisz znowu RUN, możesz stracić wszystkie wartości tak starannie wprowadzonych przez Ciebie zmiennych!

Oto co możemy zrobić zamiast tego. Gdy program zakończy przyjmowanie nowego nazwiska, dokonamy programowego sprawdzenia czy cokolwiek zostało wprowadzone jako wartość NAME$(x), innymi słowy sprawdzenie czy jako wartość NAME$(x) wprowadziliśmy

ciąg pusty czy też nie. Jeżeli (IF) tak będzie, to (THEN) zatrzymamy program bez żadnego dalszego przyjmowania danych. Czy zrozumiałeś w jaki sposób możemy to zrobić? Dodaj:

```
55 IF NAME$(x)=" " THEN 80
80 PRINT "koniec wprowadzania"
```

A także, aby program podpowiadał Ci w jaki sposób zakończyć wprowadzanie, dodaj:

```
47 PRINT "Sam [RETURN] kończy wprowadzanie"
```

A teraz napiszmy kawałek programu do wyświetlania przechowywanej informacji, początkowo w formie listy. Dodaj:

```
90 FOR x=1 TO 100
100 PRINT NAME$(x);" "; TEL$(x)
110 NEXT
```

I znowu program nie wie kiedy skończyć przed osiągnięciem 100-go elementu tablicy, a zatem dodaj jeszcze:

```
95 IF NAME$(x)=" " THEN 120
120 PRINT "lista skończona"
```

Linia 95 wykrywa czy NAME$(x) jest pustym ciągiem i gdy (IF) tak jest, to (THEN) przerywa wyświetlanie przez pominięcie linii 100 i 110.

A teraz następne nasze wymaganie. Napiszemy program który wyszukuje określone podane przez Ciebie nazwisko. Dodaj:

```
130 INPUT "znajdź"; SEARCH$
140 FOR x=1 TO 100
150 IF INSTR(NAME$(x),SEARCH$)=Ø THEN 180
160 PRINT NAME$(x);" ";TEL$(x)
170 END
180 NEXT
190 PRINT "nazwiska nie znaleziono"
run
```

W linii 150 występuje nowa instrukcja - INSTR. Powoduje ona przeszukiwanie (INterrogate) pierwszego ciągu tekstowego (STring) w celu stwierdzenia, czy występuje w nim drugie wyrażenie tekstowe, a zatem szuka się w NAME$(x) zmiennej SEARCH$ (która zawiera wprowadzone przez Ciebie w linii 130 nazwisko

do znalezienia). Jeżeli INSTR nie znajdzie zadanej zmiennej (lub żadnej części tej zmiennej), przyjmuje wartość 0, co jest tu wykorzystywane do przejścia do linii 180 i ponowienia próby z następną (NEXT) wartością x. Jeżeli program przejdzie przez wszystkie wartości x aż do 100, wykonywana jest linia 190, w wyniku czego dowiesz się że nazwiska nie znaleziono. Jeżeli jednak znalezione zostanie nazwisko, INSTR nie stanie się równe 0 i program przejdzie z linii 150 do 160, wydrukuje nazwisko i numer telefonu i zakończy się w linii 170 (END).

Jak widzisz, nasz program opracowaliśmy całkiem szybko, lecz jest jeszcze ciągle wiele do zrobienia. Usiądź przez chwilę z założonymi do tyłu rękami i rozważ niektóre cechy ujemne programu, zaczynając od sposobu w jaki program działa: najpierw Ty piszesz informacje, potem otrzymujesz z powrotem listę i wreszcie szukasz określonego nazwiska.

Co robić jeśli ...?

Dobrze, a co robić jeśli nie chcesz wykonywać programu w takiej kolejności? Co robić jeśli chcesz rozpocząć od wyszukania nazwiska, które wpisałeś do programu wczoraj? I co - jeśli chcesz dodać jeszcze więcej nazwisk i numerów do tych, które już tam są? Są to problemy o których musisz pomyśleć i znaleźć ich rozwiązanie; i to wszystko musisz zaprogramować. Jak poprzednio wspomniano, BASIC jest na tyle miły, że pozwala Ci wprowadzać wstawki między linie istniejącego programu, lecz dobry programista powinien przewidywać takie problemy z góry.

Innym ważnym problemem, związanym z tym programem jest to, że wartości zmiennych w tablicach są przechowywane wyłącznie w części pamięci komputera, która jest czyszczona za każdym razem, kiedy uruchamiasz (RUN) jakiś program. Nie chciałbyś musieć wpisywać wszystkich informacji za każdym razem, kiedy używasz programu, potrzebujesz zatem takiego rozwiązania, które umożliwiałoby zachowanie wszystkich wartości NAME$ i TEL$ nawet gdy wyłączysz komputer a następnie wprowadzić te wartości gdy znów uruchomisz program.

Rozwiązania

Pierwszy z tych problemów (tj. kolejność w której wykonywane są poszczególne czynności) może być rozwiązany przez napisanie programu, który po uruchomieniu umożliwia wybór różnych funkcji, jakie może wykonywać. Taki typ programu jest nazywany "menu-driven", tj. "sterowany na podstawie listy poleceń" i wyświetla na ekranie "menu" z którego możesz wybrać jakąś opcję. Jeżeli kiedyś korzystałeś z automatu bankowego, realizującego czeki, to musiałeś już się spotkać z programem komputerowym tego typu. A zatem dodajmy menu do naszego programu:

```
32 PRINT "1. w p r o w a d ź   i n f o"
33 PRINT "2. l i s t u j   i n f o"
34 PRINT "3. s z u k a j"
35 PRINT "4. p r z e c h o w a j   i n f o"
36 PRINT "5. ł a d u j   i n f o"
37 INPUT "w y b i e r z   o p c j ę"; ms
38 ON ms GOSUB 40,90,130

85 RETURN
125 RETURN
170 RETURN
200 RETURN
```

Jak widzisz, wprowadziliśmy program wyświetlający (PRINT) listę opcji, który następnie przyjmuje (INPUT) Twój wybór i podstawia odpowiednią wartość pod zmienną ms (skrót od menu selection). Instrukcja ON ms GOSUB w linii 38 steruje dalszym przebiegiem programu: gdy ms=1 to GO (idź) do pierwszego podprogramu (SUB-routine), rozpoczynającego się w linii (40); gdy ms=2 to GO (idź) do drugiego podprogramu w linii (90) itd.

Ponieważ każda z funkcji jest teraz podprogramem wywoływanym przez instrukcję ON ms GOSUB, a każdy podprogram musi mieć na końcu instrukcję RETURN, dodaliśmy jeszcze wszystkie te instrukcje.

Czy pamiętasz co wykonuje rozkaz RETURN? Nakazuje wrócić po wykonaniu podprogramu do miejsca w programie, występującego bezpośrednio po wywołującej podprogram instrukcji GOSUB tak, że w tym przypadku powrót nastąpi do instrukcji po linii 38.

Oznacza to, że program będzie kontynuowany od linii 40 - gdzie rozpoczyna się "wprowadzanie informacji"! Ponieważ nie chcemy, aby to nastąpiło, musimy jeszcze dodać:

```
39 GOTO 32
```

przez co zainstalujemy w programie pętlę, powodującą ponowne wyświetlenie menu. A teraz uruchom /RUN/ program znowu i zobacz jak daleko się posunęliśmy.

Spójrzmy następnie przez chwilę na listę programu. /Jeżeli program ciągle działa, naciśnij dwa razy ESC .

Wprowadź:

```
LIST
```

To, co powinieneś mieć w tym momencie przedstawiono poniżej:

```
10 REM ksiazka telefoniczna
20 DIM NAME$(100)
30 DIM TEL$(100)
32 PRINT "1. wprowadz info"
33 PRINT "2. listuj info"
34 PRINT "3. szukaj"
35 PRINT "4. przechowaj info"
36 PRINT "5. laduj info"
37 INPUT "wybierz opcje";ms
38 ON ms GOSUB 40,90,130
39 GOTO 32
40 FOR x=1 TO 100
45 CLS
47 PRINT "Sam [RETURN] konczy wprowadzanie"
50 INPUT;"nazwisko";NAME$(x)
55 IF NAME$(x)="" THEN 80
60 INPUT "telefon";TEL$(x)
70 NEXT
80 PRINT "koniec wprowadzania"
85 RETURN
90 FOR x=1 TO 100
95 IF NAME$(x)="" THEN 120
100 PRINT NAME$(x);" ";TEL$(x)
110 NEXT
120 PRINT "lista skonczona"
125 RETURN
130 INPUT "znajdz";SEARCH$
140 FOR x=1 TO 100
150 IF INSTR(NAME$(x),SEARCH$)=0 THEN 180
160 PRINT NAME$(x);" ";TEL$(x)
170 RETURN
180 NEXT
190 PRINT "nazwiska nie znaleziono"
200 RETURN
```

Można zauważyć, że w niektórych częściach programu straciliśmy możliwość wprowadzenia dodatkowych linii, wytwórzmy więc nieco więcej miejsca i uporządkujmy program przez "PRZENUMEROWANIE" linii za pomocą instrukcji RENUM. Wprowadź:

```
RENUM
LIST
```

Powinieneń zobaczyć teraz następującą listę programu:

```
10 REM ksiazka telefoniczna
20 DIM NAME$(100)
30 DIM TEL$(100)
40 PRINT "1. wprowadz info"
50 PRINT "2. listuj info"
60 PRINT "3. szukaj"
70 PRINT "4. przechowaj info"
80 PRINT "5. laduj info"
90 INPUT "wybierz opcje";ms
100 ON ms GOSUB 120,210,270
110 GOTO 40
120 FOR x=1 TO 100
130 CLS
140 PRINT "Sam [RETURN] konczy wprowadzanie"
150 INPUT;"nazwisko";NAME$(x)
160 IF NAME$(x)="" THEN 190
170 INPUT "telefon";TEL$(x)
180 NEXT
190 PRINT "koniec wprowadzania"
200 RETURN
210 FOR x=1 TO 100
220 IF NAME$(x)="" THEN 250
230 PRINT NAME$(x);" ";TEL$(x)
240 NEXT
250 PRINT "lista skonczona"
260 RETURN
270 INPUT "znajdz";SEARCH$
280 FOR x=1 TO 100
290 IF INSTR(NAME$(x),SEARCH$)=0 THEN 320
300 PRINT NAME$(x);" ";TEL$(x)
310 RETURN
320 NEXT
330 PRINT "nazwiska nie znaleziono"
340 RETURN
```

Jest lepiej. A zatem dalej z programem! Dodamy teraz instrukcję, która zapewni, że ilekroć wprowadzisz nowe nazwisko i numery telefonów, komputer dołączy je do wcześniej wprowadzonych przez umieszczenie ich w pierwszym pustym elemencie tablicy. Użyjemy w tym celu nowego rozkazu LEN, który określi długość /LENght/ ciągu. Sformułujemy to nstępująco:

Gdy /IF/ długość /LENght/ zmiennej tekstowej NAMES jest większa od 0, tzn. są już wprowadzone jakieś dane do badanego elementu tablicy, to /THEN/ skocz do linii 180 /co powoduje przeskoczenie do nstępnego /NEXT/ elementu tablicy/.

Zauważ /jeśli możesz/ jak podobna jest powyższa instrukcja sformułowana w języku angielskim do umieszczonej poniżej odpowiadającej jej instrukcji w języku BASIC. Mówię Ci, że BASIC naprawdę nie jest sztucznym językiem!!!

```
135 IF LEN/NAMES/x//  Ø THEN 180
```

Jakie to proste rozwiązanie, nieprawda? Problemy podobne do tego zawsze mogą być rozpracowane, jeżeli pomyśli się trochę nad listą słów kluczowych języka BASIC. Znajdzie się w niej prawie zawsze co najmniej jeden rozkaz, który będzie spełniał Twoje potrzeby programowe i im więcej będziesz programował tym łatwiej będziesz mógł znaleźć natychmiastowe rozwiązanie bez mała "na czubku głowy".

A teraz do sprawy przechowywania zawartości zmiennych tak, abyś mógł je wprowadzać z powrotem po ponownym uruchomieniu programu. W części 7 "kursu podstawowego" wyjaśniono jak przechować sam program, używając rozkazu SAVE. Jednakże program jest tylko szkieletem który umożliwia wprowadzanie (za pomocą klawiatury) i wyprowadzanie (na ekran) wartości zmiennych. Jeżeli przechowasz (SAVE) program, to przechowasz szkielet a nie wartości zmiennych.

A zatem musimy napisać odcinek programu, który umożliwi przechowanie wartości zmiennych na dysku. Zrobimy to przez utworzenie oddzielnego "zbioru danych".

Najpierw otworzymy (OPEN) zbiór wyjściowy (OUTput file) i przydzielimy mu nazwę "dane" ("data"). Następnie wpiszemy (WRITE) do tego zbioru wartości zmiennych NAME$(X) i TEL$(X) dla X od 1 do 100 i na koniec zamkniemy (CLOSE) nasz zbiór wyjściowy (OUTput file) i powrócimy (RETURN) do menu. Dodajmy to do programu, zaczynając od linii 350. Aby uniknąć pisania za każdym razem numeru linii użyjmy polecenia:

```
AUTO 350
```

które wprowadzi AUTOmatyczne numerowanie linii od wymienionej linii:

```
350 OUPENOUT "dane"
360 FOR x=1 TO 100
370 WRITE #9, NAME$(x),TEL$(x)
380 NEXT
390 CLOSEOUT
400 PRINT "dane przechowane"
410 RETURN
```

Po napisaniu linii 410 i naciśnięciu RETURN , naciśnij także ESC aby zakończyć AUTOmatyczne numerowanie linii.

Teraz musimy dodać jeszcze jeden numer do listy numerów linii w instrukcji ON ms GOSUB w linii 100, ponieważ wprowadziliśmy możliwość wybrania jeszcze jednej opcji z menu. A zatem popraw (EDIT) linię 100 tak aby dodać ten numer:

100 ON ms GOSUB 120,210,270,350

Teraz, gdy tylko wybierzesz z menu opcję 4, program zapisze na dysk wszystkie wprowadzone informacje.

Zauważ że w linii 370, gdzie program zapisuje (WRITE) wartości NAME$(x) i TEL$(x) na dysk, po WRITE jest umieszczone wyrażenie #9. Znak # jest "wyróżnikiem strumienia" a następujący po nim parametr określa który "strumień ma być użyty do przesyłania informacji. Komputer ma 10 takich strumieni.

Skierowanie danych do strumieni 0 do 7 (#0 do #7) powoduje pojawienie się ich na ekranie, ponieważ strumienie #Ø do #7 są "strumieniami ekranowymi" czyli "oknami ekranu" (WINDOWs).

Skierowanie danych do strumienia #8 wysyła dane do drukarki (jeżeli jest dołączona).

I wreszcie, skierowanie danych do strumienia #9 wysyła je do napędu dyskowego, co zrobiliśmy właśnie w linii 37Ø.

Zróbmy krótką dygresję ...

Dodajmy jeszcze kilka słów o rozkazie AUTO, którego używaliśmy przed chwilą. Jeżeli opuścisz w tym rozkazie numer linii i po prostu napiszesz:

AUTO

komputer zacznie numerować linie począwszy od linii 10, zwiększając numer linii o 10 po każdym naciśnięciu klawisza RETURN . Jeżeli wcześniej używałeś linii 10, 20, 30 itd., treść każdej linii wyświetlona zostanie na ekranie w miarę przechodzenia przez te linie po kolejnym naciśnięciu RETURN . Każda linia wyświetlona na ekranie może być zmieniona (poddana zabiegom edycyjnym) przed użyciem klawisza RETURN co stanowi szybką metodę wprowadzania poprawek w następujących po sobie w regularnych odstępach liniach programu.

Wracamy do programu ...

Mamy już dodane instrukcje umożliwiające przechowanie informacji na dysku tak, że ostatnią brakującą częścią programu jest wprowadzanie informacji z powrotem z dysku w taki sposób, aby były gotowe do użycia. A zatem musimy jeszcze dodać jedną opcję menu do listy numerów w linii 100. Zmień tę linię jak następuje:

```
100 ON ms GOSUB 120,210,270,350,420
```

A teraz instrukcje do wprowadzania informacji. Zaczniemy od otwarcia (OPEN) dyskowego zbioru wejściowego (INput file) o nazwie "dane". Następnie pobierzemy (INPUT) z dysku (strumień #9) wszystkie wartości zmiennych NAME$(X) i TEL$(X) dla x od 1 do 100 i na koniec zamkniemy (CLOSE) zbiór wejściowy (INput file) i powrócimy (RETURN) do menu. Napisz:

```
420 OPENIN "dane"
430 FOR x=1 TO 100
440 INPUT #9, NAMES(x), TELS(x)
450 NEXT
460 CLOSEIN
470 PRINT "dane wprowadzone"
480 RETURN
```

Koniec początku ...

Teraz mamy już napisany program który całkowicie spełnia wymagania jakie sformułowaliśmy gdy decydowaliśmy "co program ma robić". Wszystko, co jeszcze zostało to udoskonalenie sposobu w jaki "rezultaty są prezentowane" na ekranie.

Początek końca ...

Dodajmy kilka instrukcji które uporządkują prezentację programu:

```
34 MODE 1
```

Ustali to tryb pracy ekranu i wyczyści ekran po rozpoczęciu działania programu. A teraz dodaj:

```
36 WINDOW #1,12,30,10,14
```

Nie denerwuj się tą na pozór skomplikowaną instrukcją. To co robimy teraz to wytworzenie małego okna na ekranie w którym wyświetlane będzie menu. Po słowie WINDOW (okno) najpierw określiliśmy numer strumienia z którym skojarzone jest to okno, a jak pamiętasz - mogliśmy użyć w tym celu jednego z 8 strumieni ekranowych od #Ø do #7. Mając na uwadze że wszystko jest wyświetlane na ekranie przy użyciu strumienia # Ø jeżeli nie zostanie postanowione inaczej, nie użyliśmy # Ø do naszego małego okienka, ponieważ wszystko co program wyświetla, będzie wysyłane do tego strumienia. Zamiast tego określiliśmy inny strumień spomiędzy #1 do # 7 i jak widzisz wybraliśmy #1. Cztery liczby następujące po # 1 określają wymiar okna w praktycznie najłatwiejszy sposób: liczby specyfikują lewy, prawy, górny i dolny brzeg okna i odnoszą się do kolumn tekstu i numerów wierszy (tak samo jak w rozkazie LOCATE). A zatem w naszym przykładzie, po określeniu użycia strumienia # 1, powiedzieliśmy że lewy skraj okna zaczyna się w kolumnie 13, prawy skraj okna kończy się w kolumnie 3Ø, górny skraj okna rozpoczyna się w wierszu 1Ø a dolny skraj okna kończy się w wierszu 14.

Aby zapewnić wyświetlanie wszystkich pozycji naszego menu w określonym w ten sposób oknie, przy użyciu strumienia #1, musimy przeredagować linie 4Ø do 8Ø jak następuje:

```
40 PRINT #1,"1. w p r o w a d ź   i n f o"
50 PRINT #1,"2. l i s t u j   i n f o"
60 PRINT #1,"3. s z u k a j"
70 PRINT #1,"4. p r z e c h o w a j   i n f o"
80 PRINT #1,"5. ł a d u j   i n f o"
```

A teraz dodaj jeszcze:

```
85 LOCATE 7,25
```

Lokuje to początek wyświetlania napisu, przez instrukcję INPUT z linii 90 tak, by to zgrabnie wyglądało.

Aby ekran był czyszczony zawsze, kiedy powraca się do menu, przeredaguj linię 110 na

```
11Ø GOTO 34
```

Ponadto, aby ekran był czyszczony zawsze gdy wybierze się którąś z opcji, dodaj:

```
95 CLS
```

Na koniec, dodaj jeszcze następujące 3 linie aby wprowadzić w programie pauzę przed powrotem do menu:

```
103 LOCATE 4,25
105 PRINT "n a c i ś n i j   d o w o l n y   k l a w i s z
    p o   m e n u"
107 IF INKEY$=" " THEN 107
```

Linia 103 lokuje początek tekstu, wyświetlanego w linii 105. Linia 107 bada (INterrogates) stan klawiatury (KEYboard) w celu stwierdzenia jaka zmienna tekstowa (String variable) została wprowadzona w wyniku naciśnięcia klawisza. Jeżeli (IF) okaże się, że ciąg tekstowy jest pusty (ponieważ nie naciśnięto żadnego klawisza) to (THEN) program wraca z powrotem ciągle do tej samej instrukcji dopóki INKEY$ nie wykryje że ciąg tekstowy nie jest pusty gdyż naciśnięto klawisz. Jest to użyteczny sposób wprowadzania pauzy w programie przez uniemożliwienie przejścia do następnej linii programu przed naciśnięciem klawisza.

I w ten sposób mamy ukończony program. Naprawdę? Oczywiście mógłbyś wzbogacać go o takie udogodnienia jak poprawianie i wymazywanie nazwisk i numerów telefonów, układanie listy w porządku alfabetycznym, umożliwienie drukowania listy przez drukarkę lub, jeśli jesteś bardzo ambitny, automatyczne wysyłanie sygnału, odpowiadającego "wykręcaniu" numeru po napisaniu nazwiska - jeżeli oczywiście uzyskasz zgodę Urzędu Telekomunikacyjnego na podłączenie komputera do telefonu!

Wszystkie takie rozszerzenia programu są możliwe i po prawdzie mógłbyś ten program rozbudowywać i udoskonalać bez końca, zwłaszcza gdy masz tak potężny system komputerowy jak 6128. Nie, nie dopiszemy już ani jednej linii i zostawimy w tym stanie naszą "książkę telefoniczną", mając nadzieję, że nauczyliśmy się kilku rzeczy o sztuce pisania programu "od jednego pociągnięcia pióra". Uporządkuj jeszcze tylko program przez wprowadzenie:

```
RENUM
```

a potem zapisz go na dysk lub go wyrzuć. Nigdy nie wiadomo, czy będzie Ci się przydawał do przechowania nazwisk przyjaciół i ich numerów telefonów!

Ostateczną listę programu przedstawiono na następnej stronie

```
10 REM ksiazka telefoniczna
20 DIM NAME$(100)
30 DIM TEL$(100)
40 MODE 1
50 WINDOW #1,13,30,10,14
60 PRINT #1,"1. wprowadz info"
70 PRINT #1,"2. listuj info"
80 PRINT #1,"3. szukaj"
90 PRINT #1,"4. przechowaj info"
100 PRINT #1,"5. laduj info"
110 LOCATE 7,25
120 INPUT "wybierz opcje";ms
130 CLS
140 ON ms GOSUB 190,290,350,430,500
150 LOCATE 9,25
160 PRINT "nacisnij dowolny klawisz po menu"
170 IF INKEY$="" THEN 170
180 GOTO 40
190 FOR x=1 TO 100
200 CLS
210 IF LEN(NAME$(x))>0 THEN 260
220 PRINT "Sam [RETURN] konczy wprowadzanie"
230 INPUT;"nazwisko";NAME$(x)
240 IF NAME$(x)="" THEN 270
250 INPUT "telefon";TEL$(x)
260 NEXT
270 PRINT "koniec wprowadzania"
280 RETURN
290 FOR x=1 TO 100
300 IF NAME$(x)="" THEN 330
310 PRINT NAME$(x);" ";TEL$(x)
320 NEXT
330 PRINT "lista skonczona"
340 RETURN
350 INPUT "znajdz";SEARCH$
360 FOR x=1 TO 100
370 IF INSTR(NAME$(x),SEARCH$)=0 THEN 400
380 PRINT NAME$(x);" ";TEL$(x)
390 RETURN
400 NEXT
410 PRINT "nazwiska nie znaleziono"
420 RETURN
430 OPENOUT "dane"
440 FOR x=1 TO 100
450 WRITE #9,NAME$(x),TEL$(x)
460 NEXT
470 CLOSEOUT
480 PRINT "dane przechowane"
490 RETURN
500 OPENIN "dane"
510 FOR x=1 TO 100
520 INPUT #9,NAME$(x),TEL$(x)
530 NEXT
540 CLOSEIN
550 PRINT "dane wprowadzone"
560 RETURN
```

ROZDZIAŁ 3
PEŁNA LISTA SŁÓW KLUCZOWYCH JĘZYKA AMSTRAD CPC 6128 BASIC

Uwaga

Jest bardzo ważne, żeby zrozumieć terminologię i sposób zapisu stosowane w tym rozdziale. Przy wyjaśnianiu składni poszczególnych rozkazów używane są różnego typu nawiasy; należy zwrócić uwagę na to, że każdy typ nawiasu ma swoje specyficzne znaczenie.

Każda część rozkazu, która nie jest ujęta w nawiasy musi być pisana w takiej właśnie postaci. Na przykład rozkaz END ma formę:

    END

co oznacza, że należy napisać w programie dosłownie END.

Jeżeli część rozkazu jest zawarta w nawiasach ostrych na przykład:

    <numer linii>

... NIE należy wpisywać nawiasów ani słów w nich zawartych. W powyższym przykładzie <numer linii> oznacza jaki typ danych musi występować w rozkazie. Na przykład:

    EDIT < numer linii>

... oznacza, że należy napisać

    EDIT 1ØØ lub EDIT 70

Nawiasy okrągłe muszą być wpisane dosłownie. Na przykład:

    COS (<wyrażenie liczbowe>)

... oznacza, że < wyrażenie liczbowe>, którego COSinus ma być obliczony, musi być ujęte w nawiasy to jest:

    PRINT COS (45)

Wreszcie nawiasy kwadratowe zawierają opcjonalne części rozkazu lub funkcji, które mogą być napisane, ale nie muszą. Na przykład

    RUN [< numer linii>]

... oznacza, że nie trzeba koniecznie dołączyć do słowa kluczowego RUN parametru < numer linii>, ale można to zrobić. Stąd rozkaz ten może być napisany:

    RUN ... albo ... RUN 200

Znaki specjalne

£ lub	&H	Przedrostek dla stałych heksadecymalnych
&X		Przedrostek dla stałych binarnych
#		Przedrostek dla numeru strumienia logicznego

Typy danych

Ciągi znaków (teksty, łańcuchy znaków) mogą mieć długość od 0 do 255 znaków. <Wyrażenie tekstowe> jest wyrażeniem, którego "wynikiem" jest tekst. Teksty mogą być dołączone do siebie przez użycie operatora +, przy czym długość tekstu wynikowego nie może być większa niż 255.

Dane liczbowe mogą być typu liczb całkowitych lub liczb rzeczywistych. Zakres liczb całkowitych jest od -32768 do 32767, liczby rzeczywiste mają do dziewięciu cyfr znaczących, ich zakres od -1.7E+38 do +1,7E+38, a najmniejsza wartość ponad zero - około 2.9E-39.

<Wyrażenie liczbowe> jest wyrażeniem, które daje wartość liczbową. Może to być po prostu liczba, może to być zmienna liczbowa, mogą to być liczby i zmienne połączone operatorami, cokolwiek co nie jest <wyrażeniem tekstowym>.

<Wyrażenie określające strumień> odpowiada <wyrażeniu liczbowemu> i identyfikuje okno ekranu, drukarkę lub dysk, gdzie dane mają być skierowane.

<lista : <pozycji> opisuje parametr składający się z listy pozycji oddzielonych przecinkami. Lista może zawierać jedną lub kilka pozycji, liczba pozycji limitowana jest długością linii.

Znaczniki typu są następujące:

% Całkowite

! Rzeczywiste (przyjmowane w przypadku braku określenia - domyślnie)

$ Tekst (String - ciąg)

Słowa kluczowe języka AMSTRAD CPC 6128 BASIC są przedstawione w tym rozdziale w następującej formie:

SŁOWO KLUCZOWE

Składnia

Przykład

Opis
Słowa kluczowe pokrewne
Słowo kluczowe może być:
ROZKAZEM : operacje wykonywane bezpośrednio
FUNKCJĄ : operacje dołączane jak argumenty w wyrażeniu
OPERATOREM: działa na argument matematyczny

Przy LISTowaniu BASIC zamienia małe litery słów kluczowych na DUŻE LITERY. W przykładach zamieszczonych w tym rozdziale użyto DUŻYCH LITER, ponieważ tak program jest LISTowany. Przy pisaniu programu należy używać małych liter. Ułatwia to zlokalizowanie błędów, ponieważ przy listowaniu błędnie napisane słowa kluczowe są w dalszym ciągu przedstawione małymi literami.

Szczegółowy opis języka AMSTRAD CPC 6128 BASIC można znaleźć w Concise BASIC specification SOFT 967.

Słowa kluczowe ...

ABS

ABS (< wyrażenie liczbowe >)

```
PRINT ABS (-67.98)
67.98
```

FUNKCJA: Wyznacza wartość bezwzględną (ABSolute) danego wyrażenia. Oznacza to, że liczby ujemne zamieniane są na dodatnie.
Słowa kluczowe pokrewne: SGN

AFTER

AFTER < opóźnienie czasowe > [, < numer licznika >] GOSUB < numer linii >

```
10 AFTER 250 GOSUB 60:CLS
20 PRINT "Zgadnij literę w 5 sekund"
30 a$=INKEY$:IF wskaźnik = 1 THEN END
40 IF a$ <> CHR$ (INT(RND*26+97)) THEN 30
50 PRINT a$;"zgadłeś. Wygrałeś"
55 SOUND 1,478:SOUND 1,358:END
60 PRINT "za późno. Wygrałem"
70 SOUND 1,2000:wskaźnik=1:RETURN
run
```

ROZKAZ: Wywołuje podprogram napisany w BASIC'u po (AFTFR) zadanym opóźnieniu czasowym. Parametr < opóźnienie czasowe > wyznacza okres czasu w jednostkach 0,02 sekundy. < numer licznika > (w zakresie 0 do 3) określa, który z czterech liczników ma być użyty. Licznik 3 ma najwyższy priorytet, licznik 0 (przyjmowany w braku określenia - domyślnie) - najniższy.

Każdy licznik może wywoływać swój podprogram.

Dalsze informacje dotyczące przerwań mogą być znalezione w drugiej części rozdziału zatytułowanego "W wolnej chwili..."
Słowa kluczowe pokrewne: EVERY, RFMAIN, RETURN

AND

argument AND argument

```
IF "piotr"<"zenek" AND "pies">"kot" THEN PRINT "dobrze"
ELSE PRINT "źle"
dobrze
IF "zenek"<"piotr" AND "kot">"pies" THEN PRINT "dobrze"
ELSE PRINT "źle"
źle
IF "piotr"<"zenek" AND "kot">"pies" THEN PRINT "dobrze"
ELSE PRINT "źle"
źle
...
PRINT 1 AND 1
 1
PRINT 0 AND 0
 0
PRINT 1 AND 0
 0
```

OPFRATOR: Realizuje iloczyn logiczny (AND - i) kolejnych bitów liczb całkowitych. Bit wyniku jest równy 1, jeżeli odpowiadające bity obydwu argumentów są równe 1. W pozostałych przypadkach bit wyniku jest 0.

Dalsze informacje dotyczące operacji logicznych mogą być znalezione w drugiej części rozdziału "W wolnej chwili..."
Słowa kluczowe pokrewne: OR, NOT, XOR

ASC

ASC (<wyrażenie tekstowe>)

    PRINT ASC ("x")
    120

FUNKCJA: Wyznacza wartość liczbową (w kodzie ASCII) pierwszego znaku <wyrażenia tekstowego>.

Słowa kluczowe pokrewne: CHR$

ATN

ATN (<wyrażenie liczbowe>)

    PRINT ATN (1)
    0.785398163

FUNKCJA: Oblicza Arcus TaNgens <wyrażenia liczbowego>. Słowa DEG i RAD mogą być użyte dla uzyskania wyniku odpowiednio w stopniach lub radianach.
Słowa kluczowe pokrewne: COS, DEG, RAD, SIN, TAN

AUTO

AUTO [<numer linii>] [, <inkrement>]

    AUTO 100, 50

ROZKAZ: AUTOmatycznie generuje numery linii. Opcjonalny parametr <numer linii> definiuje, który numer ma być wygenerowany jako pierwszy i stosowany jest w przypadku gdy chcemy rozpocząć generację linii od jakiegoś miejsca programu. W przypadku pominięcia tego parametru numery linii generowane są od linii 10.

Opcjonalny parametr <inkrement> definiuje różnicę między kolejnymi generowanymi numerami. W przypadku pominięcia tego parametru numery kolejnych linii zwiększane są o 10.

Jeżeli generowany jest numer linii aktualnie istniejącej, treść tej linii jest wyświetlana na ekranie i może być zmieniana jeżeli trzeba. Treść wyświetlanej linii jest zmieniana w pamięci w chwili naciśnięcia [RETURN].

Aby przerwać automatyczne numerowanie linii, należy nacisnąć [ESC].

Słowa kluczowe pokrewne: nie ma

BIN$

BIN$ (<wyrażenie określające liczbę całkowitą> [,<wyrażenie określające liczbę całkowitą>])

```
PRINT BIN$ (64,8)
0 1 0 0 0 0 0 0
```

FUNKCJA: Wyznacza ciąg cyfr BINarnych reprezentujących wartość pierwszego <wyrażenia określającego liczbę całkowitą> , przy czym liczba cyfr binarnych jest równa drugiemu <wyrażeniu określającemu liczbę całkowitą> (w zakresie 0 do 16). Jeżeli żądana liczba cyfr jest za duża, wynik jest zapełniany nieznaczącymi zerami; jeżeli żądana liczba cyfr jest za mała, wynik NIE jest skracany do tej długości, lecz jest podawany z taką liczbą cyfr jaka jest potrzebna.

Pierwsze <wyrażenie określające liczbę całkowitą> , które ma być przedstawione w postaci binarnej, musi dawać wartość w zakresie -32768 do 65535.

Słowa kluczowe pokrewne: DEC$, HEX$, STR$

BORDER

BORDER <kolor> [,<kolor>]

```
10 REM 729 kombinacji koloru obrzeża
20 SPEED INK 5,5
30 FOR a=0 TO 26
40 FOR b=0 TO 26
50 BORDER a,b:CLS:LOCATE 14,13
60 PRINT "border"; a; ","; b
70 FOR t=1 TO 500
80 NEXT t,b,a
run
```

ROZKAZ: Zmienia kolor obrzeża (BORDER) ekranu. Jeżeli podane są dwa kolory, obrzeże ekranu zmienia kolor co czas ustalany rozkazem SPEED INK. Zakres numerów kolorów jest od 0 do 26.

Słowa kluczowe pokrewne: SPEED INK

BREAK

(zobacz ON BREAK CONT, ON BREAK GOSUB, ON BREAK STOP)

CALL

CALL < wyrażenie określające adres > [,< lista: < parametrów >]

    CALL 0

ROZKAZ: Pozwala na wywoływanie (CALL) w BASIC'u podprogramów zewnętrznych dla BASIC'a. Powyższy przykład całkowicie zeruje komputer.

Rozkaz ten nie powinien być stosowany przez niedoświadczonego użytkownika.

Słowa kluczowe pokrewne: UNT

CAT

    CAT

ROZKAZ: Podaje katalog (CATaloque) dysku. Wyświetla w porządku alfabetycznym pełne nazwy zbiorów razem z ich długością (w zaokrągleniu w górę do pełnego kilobajtu). Wyświetlana jest też ilość wolnego miejsca na dysku oraz identyfikacja napędu dysku (Drive) i użytkownika (User).

Rozkaz ten nie niszczy programu aktualnie znajdującego się w pamięci.

Słowa kluczowe pokrewne: LOAD, RUN, SAVE

CHAIN

CHAIN < nazwa zbioru > [, < wyrażenie określające numer linii >]

    CHAIN "testprog.bas", 350

ROZKAZ: Ładuje program z dysku do pamięci w miejsce aktualnie znajdującego się w pamięci programu. Następnie nowy program zaczyna być wykonywany od początku lub od linii podanej w opcjonalnym < wyrażeniu określającym numer linii > .

Zbiory zabezpieczone (z przyrostkiem , p) mogą być ładowane i uruchamiane rozkazem CHAIN.

Słowa kluczowe pokrewne: CHAIN MERGE, LOAD, MERGE

CHAIN MERGE

CHAIN MERGE <nazwa zbioru> [, <wyrażenie określające numer linii>][, DELETE <zakres numerów linii>]

```
CHAIN MERGE "newrun.bas", 750, DELETE 400-680
```

ROZKAZ: Ładuje program z dysku, dołącza (MERGE) go do programu znajdującego się w pamięci, następnie rozpoczyna jego wykonywanie od początku lub od numeru linii podanego w opcjonalnym <wyrażeniu określającym numer linii> . Jeżeli, przed dołączeniem programu z dysku, należy usunąć (DELETE) część programu znajdującego się w pamięci, można dodać do rozkazu opcjonalną część ,DELETE <zakres numerów linii> .

Uwaga, linie programu pierwotnego, które występują również w programie dołączanym, są zastępowane przez linie nowego programu.

Zbiory zabezpieczone (z przyrostkiem ,p) nie mogą być ładowane i uruchamiane przez CHAIN MERGE.

Słowa kluczowe pokrewne: CHAIN, DELETE, LOAD, MERGE

CHR$

CHR$ (<wyrażenie określające liczbę całkowitą>)

```
10 FOR x = 32 TO 255
20 PRINT  x; CHR$ (x)
30 NEXT
run
```

FUNKCJA: Zamienia wyrażenie określające liczbę całkowitą w zakresie 0 do 255 na równoważny znak (CHaRacter $tring) według wykazu znaków AMSTRAD'a zamieszczonego w trzeciej części rozdziału zatytułowanego "Nieco użytecznych informacji ..."

Uwaga, liczby od 0 do 31 odpowiadają znakom sterującym; stąd w powyższym przykładzie wypisane są znaki w zakresie 32 do 255.

Słowa kluczowe pokrewne: ASC

CINT

CINT (<wyrażenie liczbowe>)

```
10 n=1,9999
20 PRINT CINT (n)
run
2
```

FUNKCJA: Zaokrągla wartość <wyrażenia liczbowego> do liczby całkowitej (Convert to INTeger) w zakresie -32768 do 32767.
Słowa kluczowe pokrewne: GREAL, FIX, INT, ROUND, UNT

CLEAR

CLEAR

```
CLEAR
```

ROZKAZ: Zeruje (CLEAR) wszystkie zmienne. Rozkaz ten powoduje zaniechanie otwartych zbiorów, skasowanie wszystkich tablic i funkcji użytkownika oraz przełączenie BASIC'a na obliczenia kątów w radianach.
Słowa kluczowe pokrewne: nie ma

CLEAR INPUT

CLEAR INPUT

```
10 CLS
20 PRINT "Wpisuj teraz litery!"
30 FOR t=1 TO 3000
40 NEXT
50 CLEAR INPUT
run
```

ROZKAZ: Usuwa z bufora klawiatury wszystkie wcześniej wpisane znaki.

Aby zaobserwować działanie tego rozkazu należy uruchomic powyższy program i wpisywać litery z klawiatury, kiedy komputer poprosi o to. Następnie usunąć linię 50 i uruchomić program jeszcze raz, zwracając uwagę na różnice w działaniu.
Słowa kluczowe pokrewne: INKEY, INKEY$, JOY

CLG

CLG [<atrament>]

```
LOCATE 1, 20
CLG 3
```

ROZKAZ: Czyści ekran graficzny (**Clear the Graphics screen**) aktualnym kolorem papieru graficznego. Jeżeli podano <atrament> , wartość ta zostaje przypisana do papieru graficznego.
Słowa kluczowe pokrewne: CLS, GRAPHICS PAPER, INK, ORIGIN

CLOSEIN

CLOSEIN

```
CLOSEIN
```

ROZKAZ: Zamyka zbiór wejściowy (CLOSE INput) z dysku. (Zobacz OPENIN).
Słowa kluczowe pokrewne: EOF, OPENIN

CLOSEOUT

CLOSFOUT

```
CLOSEOUT
```

ROZKAZ: Zamyka zbiór wyjściowy (CLOSE OUTput) do dysku. (Zobacz OPENOUT).
Słowa kluczowe pokrewne: OPENOUT

CLS

CLS [#<wyrażenie określające strumień>]

```
10 PAPER 2,3
20 CLS 2
run
```

ROZKAZ: Czyści daną część ekranu (Clear Screen) (okno) do koloru papieru. Jeżeli nie podano <wyrażenia określającego strumień>, czyszczony jest strumień #0 (cały ekran).
Słowa kluczowe pokrewne: CLG, INK, PAPER, WINDOW

CONT

CONT

CONT

ROZKAZ: Kontynuuje (CONTinue) wykonywanie programu jeżeli klawisz [ESC] został naciśnięty dwukrotnie lub jeżeli w programie zamieszczono rozkaz STOP. CONT kontynuuje wykonywanie programu, jeżeli nie był on zmieniony i jeżeli nie jest to program zabezpieczony.

Przed kontynuacją można wprowadzać inne rozkazy bezpośrednie.

Słowa kluczowe pokrewne: STOP

COPYCHR$

COPYCHR$ (#<wyrażenie określające strumień>)

```
10 CLS
20 PRINT "górny róg"
30 LOCATE 1,1
40 a$ = COPYCHR$ (#0)
50 LOCATE 1,20
60 PRINT a$
run
```

FUNKCJA: Kopiuje znak (COPY CHaRacter) z danej pozycji w strumieniu (który MUSI być podany). Powyższy program kopiuje znak z pozycji 1,1 (z lewej strony u góry) ekranu i przepisuje go na pozycję 1,20

Jeżeli znak nie jest rozpoznany wynik jest ciągiem pustym.

Słowa kluczowe pokrewne: LOCATE

COS

COS (<wyrażenie liczbowe>)

```
DEG
PRINT COS (45)
0.707106781
```

FUNKCJA: Oblicza COSinus <wyrażenia liczbowego>.

Rozkazy DEG i RAD powodują, że wartość kąta podstawiana jest odpowiednio w stopniach lub radianach.
Słowa kluczowe pokrewne: ATN, DEG, RAD, SIN

CREAL

CREAL (<wyrażenie liczbowe>)

```
10 a = PI
20 PRINT CINT (a)
30 PRINT CREAL (a)
run
3
3.14159265
```

FUNKCJA: Oblicza wartość <wyrażenia liczbowego>. Przekształca ją na postać rzeczywistą (Convert to REAL).
Słowa kluczowe pokrewne: CINT

CURSOR

CURSOR [<przełącznik systemowy>] [, <przełącznik użytkownika>]

```
10 CURSOR 1
20 PRINT "pytanie?"
30 a$ = INKEY$ : IF a$ = "" THEN 30
40 PRINT a$
50 CURSOR 0
run
```

ROZKAZ: Włącza lub wyłącza przełączniki, systemowy i użytkownika, kursora. Obydwa parametry muszą mieć wartość 0 (wyłączone) lub 1 (włączone). W powyższym rozkazie INKEY$, gdzie kursor jest normalnie wyłączony, został on włączony dzięki istnieniu linii 10.

Kursor jest wyświetlany, kiedy zarówno <przełącznik systemowy>, jak i <przełącznik użytkownika> są włączone (1). <przełącznik systemowy> jest automatycznie włączany dla rozkazu INPUT, ale wyłączany dla rozkazu INKEY$.

Poleca się, żeby kursor był wyłączony przy pisaniu tekstu na ekranie.

Każdy parametr przełącznikowy może być pominięty, ale nie obydwa jednocześnie. Jeżeli jeden parametr jest pominięty, dany przełącznik nie zmienia swego stanu.
Słowa kluczowe pokrewne: LOCATE

DATA

DATA <lista:<stałych>

```
10 FOR x = 1 TO 4
20 READ imię$, nazwisko$
30 PRINT imię$; " "; nazwisko$
40 NEXT
50 DATA Hilda, Ogden, Bet, Lynch
60 DATA Rita, Fairclough, Mavis, Riley
run
```

ROZKAZ: Deklaruje dane (DATA) do użytku w programie. Dane te mogą być wczytywane do zmiennych za pomocą rozkazu READ, po którym "wskaźnik" przesuwa się do następnej pozycji w liście danych. Rozkaz RESTORE umożliwia przesuwanie wskaźnika do żądanej pozycji.

Dalsze informacje dotyczące danych mogą być znalezione w części 2 rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: READ, RESTORE

DEC$

DEC$ (<wyrażenie liczbowe>, <szablon formatowy>)

```
PRINT DEC$ (10↑7,"££#########,.##")
£10,000,000.00
```

FUNKCJA: Wytwarza ciąg znaków dziesiętnych (DECimal $tring) reprezentujących wartość <wyrażenia liczbowego>, używając podanego <szablonu formatowego> do sterowania formatem wydruku ciągu wynikowego.
Szablon formatowy może zawierać wyłącznie znaki:
+ -£$*#,.↑
Użycie tych "specyfikatorów pola formatu" jest opisane pod słowem kluczowym PRINT USING.
Słowa kluczowe pokrewne: BIN$, HEX$, PRINT USING, STR$

DEF FN

DEF FN <nazwa funkcji> [(< parametry formalne>)] = <wyrażenie>

```
10 t = TIME/300
20 DEF FNzegar = INT (TIME/300-t)
30 EVERY 100 GOSUB 50
40 GOTO 40
50 PRINT "program został uruchomiony";
60 PRINT FNzegar; "sekund temu"
70 RETURN
run
```

ROZKAZ: DEFiniuje FuNkcję. BASIC pozwala na definiowanie i używanie w programie funkcji jednowyjściowych. DEF FN jest częścią definicyjną tego mechanizmu i tworzy funkcję programową, która działa w programie tak samo jak na przykład wbudowana na stałe funkcja COS.
(Uwaga, w powyższym przykładzie wartość funkcji FN zegar jest stale uaktualniana, nawet, jeżeli program został przerwany przez jednokrotne naciśnięcie [ESC], lub zatrzymany przez dwukrotne naciśnięcie [ESC], a następnie kontynuowany (CONT)).
Słowa kluczowe pokrewne: nie ma

DEFINT

DEFINT <lista:<zakresów liter>

```
10 DEFINT n
20 liczba = 123.456
30 PRINT liczba
run
123
```

ROZKAZ: Ustala typ zmiennej, wobec braku innego określenia (DEFault), na typ "całkowite" (INTeger). Typ ten jest stosowany wtedy, kiedy zmienna jest wymieniana bez jawnego znacznika typu (!, %, $). Rozkaz ten ustala typ "całkowite" dla zmiennych, których pierwsze litery są takie same, jak wyszczególnione w <liście>. Lista pierwszych liter może być na przykład:

```
DEFINT a, b, c
```

... lub w postaci zamkniętego przedziału pierwszych liter:

```
DEFINT a-z
```

Słowa kluczowe pokrewne: DEFREAL, DEFSTR

DEFREAL

DEFREAL <lista: <zakresów liter>

```
DEFREAL x,a-f
```

ROZKAZ: Ustala typ zmiennej, wobec braku innego określenia (DEFault), na typ "rzeczywiste" (REAL). Typ ten jest stosowany wtedy, kiedy zmienna jest wymieniona bez jawnego znacznika typu (! % $). Rozkaz ten ustala typ "rzeczywiste" dla zmiennych, których pierwsze litery są takie same, jak wyszczególnione w < liście >. Lista pierwszych liter może być na przykład:

```
DEFERAL a,b,c
```

... lub w postaci zamkniętego przedziału pierwszych liter

```
DEFREAL a-z
```

Słowa kluczowe pokrewne: DEFINT, DEFSTR

DEFSTR

DEFSTR <lista: <zakresów liter>

```
10 DEFSTR n
20 nazwa = "Amstrad"
30 PRINT nazwa
run
Amstrad
```

ROZKAZ: Ustala typ zmiennej, wobec braku innego określenia (DEFault), na typ "tekst" (STRing). Typ ten jest stosowany wtedy, kiedy zmienna jest wymieniana bez jawnego znacznika typu (! % $). Rozkaz ten ustala typ "tekst" dla zmiennych, których pierwsze litery są takie same, jak wyszczególnione w < liście >. Lista pierwszych liter może być na przykład

```
DEFSTR a,b,c
```

... lub w postaci zamkniętego przedziału pierwszych liter:

```
DEFSTR a-z
```

Słowa kluczowe pokrewne: DEFINT, DEFREAL

DEG

DEG

    DEG

ROZKAZ: Ustala sposób liczenia kątów w stopniach (DEGrees). Początkowo komputer jest ustawiony na liczenie kątów funkcji SIN, COS, TAN i ATAN w radianach, DEG przestawia BASIC na stopnie aż do przełączenia ponownego rozkazami RAD oraz NEW, CLEAR, LOAD, RUN itp.

Słowa kluczowe pokrewne: ATN, COS, RAD, SIN, TAN

DELETE

DELETE < zakres numerów linii >

    DELETE 100-200

ROZKAZ: Usuwa (DELETE) część programu bieżącego według wyrażenia < zakres numerów linii >.

Pierwszy lub ostatni numer w <zakresie numerów linii> może być pominięty dla wskazania "... od początku programu" lub "... do końca programu", to jest:

    DELETE -200

... lub ...

    DELETE 50-

... lub ...

    DELETE

... co usuwa cały program

Słowa kluczowe pokrewne: CHAIN MERGE, RENUM

DERR

DERR

```
LOAD "xyz.abc"
XYZ.ABC not found
Ready
PRINT DERR
146
```

FUNKCJA: Podaje kod ostatniego błędu sygnalizowanego przez system dyskowy. Wartość DERR może być używana do określenia zaistniałego błędu dysku (Disc.ERRor). Zobacz spis komunikatów błędów, podany w rozdziale zatytułowanym "Nieco użytecznych informacji . ."

Słowa kluczowe pokrewne: ERL, ERR, ERROR, ON ERROR GOTO, RESUME

DI

DI

```
10 CLS:TAG:FVERY 10 GOSUB 90
20 X1=RND*320:X2=RND*320
30 Y=200+RND*200:C$=CHR$ (RND*255)
40 FOR X=320-X1 TO 320+X2 STEP 4
50 DI
60 MOVE 320,0,1:MOVE X-2, Y:MOVE X,Y
70 PRINT " "; C$;:FRAMF
80 EI:NEXT:GOTO 20
90 MOVE 320,0:DRAW X+8, Y-16, 0:RETURN
run
```

ROZKAZ: Zakaz przerwań (Disable Interupts) (innych niż przez [ESC]) aż do ponownego zezwolenia jawnie przez rozkaz EI lub niejawnie przez RETURN na końcu podprogramu obsługującego przerwanie.

Uwaga, wejście w podprogram przerwania automatycznie zakazuje przerwania o równym i niższym priorytecie.

Rozkaz ten jest stosowany, aby zapobiec przerwaniom w czasie wykonywania danego fragmentu programu - na przykład kiedy dwie części programu korzystają z jednego zbioru danych. W powyższym przykładzie program główny i podprogram obsługi przerwania korzystają z ekranu graficznego.

Dalsze informacje dotyczące przerwań można znaleźć w drugiej części rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: AFTER, EI, EVERY, REMAIN

DIM

DIM <lista:<zmiennych indeksowanych>

```
10 CLS
20 DIM koledzy$(5), telefony$(5)
30 FOR n = 1 TO 5
40 PRINT "Kolega nr."; n
50 INPUT "Wpisz nazwisko"; koledzy$ (n)
60 INPUT "Wpisz nr. telefonu"; telefony$ (n)
70 PRINT
80 NEXT
90 FOR n=1 TO 5
100 PRINT n; koledzy$(n), telefony$(n)
110 NEXT
run
```

ROZKAZ: Ustala wymiar (DIMension) tablicy. DIM przeznacza pewien obszar pamięci na tablicę i określa maksymalną liczbę zmiennych indeksowanych. BASIC musi być powiadomiony jak duży obszar pamięci ma przeznaczyć na tablicę; wobec braku określenia długość tablicy jest 10.

Tablica jest identyfikowana przez <zmienną indeksowaną> dla której używana jest jedna nazwa zmiennej z różnymi indeksami (z zakresu podanego w rozkazie DIM) tak, że każdy "element" tablicy ma swoją indywidualną wartość. "Obróbka" tablicy może być więc dokonana na przykład przez pętle FOR NEXT, które przechodząc przez kolejne elementy, zapewniają ich przekształcenie.

Uwaga, najniższa wartość indeksu jest zero (pierwszy element tablicy).

Tablice mogą być wielowymiarowe; każdy element takiej tablicy powinien być opisywany odpowiednią liczbą indeksów. Na przykład dla tablicy określonej przez:

```
DIM position$| (20, 20, 20)
```

... element powinien być w postaci, na przykład:

```
position$| (4, 5, 6)
```

Słowa kluczowe pokrewne: ERASE

DRAW

DRAW <współrzędna x>, <współrzędna y> [,[<atrament>] [,<tryb rysowania>]]

```
10 MODE 0:BORDER 0: PAPER 0:INK 0,0
20 x=RND*640:y=RND*400:z=RND*15
30 DRAW x,y,z
40 GOTO 20
run
```

ROZKAZ: Rysuje (DRAW) linię na ekranie graficznym od bieżącej pozycji kursora graficznego do pozycji określonej wartościami współrzędnych x, y. Kolor, którym ma być rysowana linia, określa opcjonalny parametr <atrament> (w zakresie 0 do 15).

<Tryb rysowania> określa charakter interakcji koloru linii rysowanej z kolorami występującymi już na ekranie graficznym. <tryby rysowania> są następujące:

0: Normalny
1: XOR (eXclusive OR - wyłącznie lub)
2: AND - i
3: OR - lub
Słowa kluczowe pokrewne: DRAWR, GRAPHICS PEN, MASK

DRAWR

DRAWR <wektor x>, <wektor y> [,[<atrament>] [,<tryb rysowania>]]

```
10 CLS:PRINT "zaczynają się schody?"
20 MOVE 0,350:FOR n=1 TO 8
30 DRAWR 50, 0
40 DRAWR 0, -50
50 NEXT:MOVE 348, 0:FILL 3
60 GOTO 60
run
```

ROZKAZ: Rysuje (DRAW) linię na ekranie graficznym od bieżącej pozycji kursora graficznego do pozycji określonej względnie (Relative) wektorem (x,y). Kolor, którym ma być rysowana linia, określa opcjonalny parametr <atrament> (w zakresie 0 do 15)

Tryb rysowania określa charakter interakcji koloru linii rysowanej z kolorami już występującymi na ekranie graficznym.

<tryby rysowania> są następujące:

0: Normalny
1: XOR (eXclusive OR - wyłącznie lub)
2: AND - i
3: OR - lub
Słowa kluczowe pokrewne: DRAW, GRAPHICS PEN, MASK

EDIT

EDIT <numer linii>

```
EDIT 20
```

ROZKAZ: Wyświetla (EDIT - edycja) linię programu podaną w rozkazie wraz z kursorem, w celu dokonania poprawek.
Słowa kluczowe pokrewne: AUTO, LIST

EI

EI

```
EI
```

ROZKAZ: Zezwala (Enable) na przerwania (Interrupts), które zostały zakazane rozkazem DI.

Jeżeli przerwania są zakazane w podprogramie obsługi przerwania, ponowne zezwolenie jest automatycznie realizowane, kiedy BASIC napotka rozkaz RETURN na końcu podprogramu.

Dalsze informacje dotyczące przerwań można znaleźć w części 2 rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: AFTER, DI, EVERY, REMAIN

ELSE

(Zobacz IF)

END

END

```
END
```

ROZKAZ: Kończy (END) wykonywanie programu, realizuje powrót do trybu bezpośredniego. W programie może występować dowolna liczba rozkazów END, jeden taki rozkaz jest automatycznie przewidywany po ostatniej linii programu.

Słowa kluczowe pokrewne: STOP

ENT

ENT <numer obwiedni>[,<sekcja obwiedni>][,<sekcja obwiedni>][,<sekcja obwiedni>][,<sekcja obwiedni>][,<sekcja obwiedni>]

```
10 ENT 1,10,-50,10,10,50,10
20 SOUND 1,500,200,10, 1
run
```

ROZKAZ: Ustala obwiednię (modulację) tonu (Tone ENvelope) przypisaną do <numeru obwiedni> (zakres 1 do 15), która jest używana w rozkazie SOUND (dźwięk). Jeżeli <numer obwiedni> jest ujemny (-1 do -15), obwiednia jest powtarzana, aż do końca czasu trwania rozkazu SOUND.

Każda z <sekcji obwiedni> może zawierać 2 lub 3 parametry. Jeżeli użyto 3 parametrów, są one następujące:
<liczba stopni> , < wartość stopnia > , < długość stopnia >

Parametr 1: < liczba stopni >

Parametr ten podaje przez ile różnych stopni tonu ma przejść dźwięk w czasie trwania sekcji obwiedni. Na przykład dla nuty, która ma trwać 10 sekund można zażądać 10 stopni tonu, każdy o czasie trwania 1 sekundy. W takim przypadku parametr <liczba stopni> powinien być 10.
Dostępny zakres <liczby stopni> jest 0 do 239.

Parametr 2: < wartość stopnia >

Parametr ten musi być w zakresie -128 do +127. Ujemne stopnie powodują, że ton jest wyższy; dodatnie powodują, że ton jest niższy. Najkrótszy okres tonu jest 0. Pełny zakres okresów tonu jest zamieszczony w rozdziale "Nieco użytecznych informacji ...".

Parametr 3: < długość stopnia >

Parametr ten określa czas pomiędzy stopniami w jednostkach 0,01 sekundowych. Zakres liczb określających ten parametr jest 0 do 255 (gdzie 0 jest traktowane jak 256), co oznacza, że najdłuższy czas między stopniami wynosi 2,56 sekundy.
Jeżeli użyto 2 parametrów, są one następujące.
< okres tonu > , < długość tonu >

Parametr 1: < okres tonu >

Parametr ten określa bezwzględnie okres tonu (patrz Parametr 2 rozkazu SOUND).

Parametr 2: < długość tonu >

Parametr ten podaje czas trwania tonu w jednostkach 0,01 sekundowych. Zakres liczb określających ten parametr jest 0 do 255 (gdzie 0 jest traktowane jak 256).

Wiadomości ogólne

Uwaga, całkowita długość wszystkich stopni powinna być nie większa niż parametr < czas trwania > w rozkazie SOUND, w przeciwnym razie dźwięk skończy się, zanim przejdzie przez wszystkie stopnie. (W tym przypadku pozostała część obwiedni tonu będzie niewykorzystana).

Podobnie, jeżeli parametr czas trwania w rozkazie SOUND jest większy niż całkowita długość wszystkich stopni, dźwięk będzie kontynuowany po przejściu wszystkich stopni, a wysokość tonu pozostanie na wartości wynikającej z ostatniego stopnia.

W rozkazie ENT można użyć do 5 różnych <sekcji obwiedni> (każda złożona z powyższych 2 lub 3 parametrów).

Pierwszy stopień obwiedni tonu jest wykonywany natychmiast.

Za każdym nowym określaniem obwiedni o danym numerze, jej poprzedni kształt przestaje obowiązywać.

Podanie numeru obwiedni bez żadnej sekcji w rozkazie ENT kasuje poprzednio ukształtowaną obwiednię.

Dalsze informacje dotyczące dźwięku można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: ENV, SOUND

ENV

ENV <numer obwiedni> [,<sekcja obwiedni>][,<sekcja obwiedni>]
[,<sekcja obwiedni>][,<sekcja obwiedni>]
[,<sekcja obwiedni>]

```
10 ENV 1,15,-1,10,15,1,10
20 SOUND 1,200,300,15,1
run
```

ROZKAZ: Ustala obwiednię (modulację) głośności (Volume ENvelope) przypisaną do <numeru obwiedni> (w zakresie 1 do 15), która jest używana w rozkazie SOUND (dźwięk).

Każda z <sekcji obwiedni> może zawierać 2 lub 3 parametry.

Jeżeli użyto 3 parametrów, są to:

<liczba stopni>, <wartość stopnia>, <długość stopnia>

Parametr 1: <liczba stopni>

Parametr ten podaje przez ile różnych stopni głośności ma przejść dźwięk w czasie trwania sekcji obwiedni. Na przykład dla nuty, która ma trwać 10 sekund można zażądać 10 stopni głośności, każdy o czasie trwania 1 sekundy. W takim przypadku parametr <liczba stopni> powinien być 10.

Dostępny zakres liczby stopni jest 1 do 127.

Parametr 2: <wartość stopnia>

Każdy stopień może różnić się od poprzedniego o poziom głośności od 0 do 15. Przy różnicy równej 16 poziom głośności jest ten sam. Wprawdzie <wartość stopnia> ma zakres -128 do +127, tym niemniej poziom głośności powtarza się co 16.

Parametr 3: <długość stopnia>

Parametr ten określa czas pomiędzy stopniami w jednostkach 0,01 sekundowych. Zakres liczb określających ten parametr jest 0 do 255 (gdzie 0 jest traktowane jak 256), co oznacza, że najdłuższy czas między stopniami wynosi 2,56 sekundy.
Jeżeli użyto 2 parametrów, są to:
<obwiednia sprzętowa>, <długość obwiedni>

Parametr 1: <obwiednia sprzętowa>

Parametr ten podaje wartość, która ma być przesłana do rejestru kształtu obwiedni układu sterowania dźwiękiem.

Parametr 2: <długość obwiedni>

Parametr ten podaje wartość, która ma być przesłana do rejestru okresu obwiedni układu sterowania dźwiękiem.

Zakłada się znajomość sprzętu przy użyciu obwiedni sprzętowej. Bez takiej znajomości, należy używać obwiedni programowych zawierających odpowiedni parametr <długość stopnia>.

Wiadomości ogólne

Uwaga, całkowita długość wszystkich stopni powinna być nie większa niż parametr <czas trwania> w rozkazie SOUND, w przeciwnym razie dźwięk skończy się zanim przejdzie przez wszystkie stopnie głośności. (W tym przypadku pozostała część obwiedni głośności będzie niewykorzystana).

Podobnie, jeżeli parametr <czas trwania> w rozkazie SOUND jest większy niż całkowita długość wszystkich stopni dźwięk będzie kontynuowany po przejściu wszystkich stopni, z głośnością wynikającą z ostatniego stopnia.

W rozkazie ENV można użyć do 5 różnych <sekcji obwiedni> (każda złożona z powyższych 2 lub 3 parametrów).
Pierwszy stopień obwiedni głośności jest wykonywany natychmiast

Za każdym nowym określeniem obwiedni o danym numerze, jej poprzedni kształt przestaje obowiązywać.

Podanie numeru obwiedni bez żadnej sekcji w rozkazie ENV kasuje poprzednio ukształtowaną obwiednię.

Dalsze informacje dotyczące dźwięku można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: ENT, SOUND

EOF

EOF

```
10 OPENIN "keys.wp"
20 WHILE NOT EOF
30 LINE INPUT # 9, a$
40 PRINT a$
50 WEND
60 CLOSEIN
run
```

FUNKCJA: Testuje czy wejście z dysku sygnalizuje koniec zbioru (End Of File). Wynik jest 1 (prawda), jeżeli nie jest otwarty żaden zbiór lub zbiór został wczytany. W przeciwnym wypadku wynik jest 0 (fałsz).
Słowa kluczowe pokrewne: OPENIN, CLOSEIN

ERASE

ERASE <lista: < nazw zmiennych >

```
DIM a (100), b$ (100)
ERASE a, b$
```

ROZKAZ: Usuwa (ERASE) zawartość tablicy już niepotrzebnej, zwalniając obszar pamięci do innego użytku.
Słowa kluczowe pokrewne: DIM

ERL

ERL

```
10 ON ERROR GOTO 30
20 GOTO 1000
30 PRINT "błąd jest w linii"; ERL
40 END
run
```

FUNKCJA: Podaje numer linii, w której został zauważony ostatni błąd (ERror Line). W powyższym przykładzie błąd sygnalizowany jest w linii 20 i jest wyświetlany przez funkcję ERL.
Słowa kluczowe pokrewne: DERR, ERR, ERROR, ON ERROR GOTO, RESUME

ERR

ERR

```
GOTO 500
Line does not exist
Ready
PRINT ERR
8
```

FUNKCJA: Podaje numer ostatnio wykrytego błędu (ERRor). Zobacz spis komunikatów błędów podanych w rozdziale zatytułowanym "Nieco użytecznych informacji ...". W powyższym przykładzie funkcja ERR podaje numer 8 odpowiadający błędowi "Linia nie istnieje".
Słowa kluczowe pokrewne: DERR, ERL, ERROR, ON ERROR GOTO, RESUME

ERROR

ERROR <wyrażenie określające liczbę całkowitą>

```
10 IF INKEY$ = " " THEN 10 ELSE ERROR 17
run
```

ROZKAZ: Wywołuje program obsługujący błąd (ERROR) o numerze podanym w <wyrażeniu określającym liczbę całkowitą> Spis komunikatów błędów 1 do 32 jest podany w rozdziale zatytułowanym "Nieco użytecznych informacji ...". BASIC traktuje ERROR, jak gdyby został wykryty prawdziwy błąd i wchodzi w podprogram obsługi błędu, jak również podaje właściwe wartości ERR i ERL.

ERROR z <wyrażeniem określającym liczbę całkowitą> w zakresie 33 do 255 może być użyty do tworzenia własnych komunikatów błędów, jak pokazano w następującym przykładzie:

```
 10 ON ERROR GOTO 100
 20 INPUT "wpisz jeden znak";a$
 30 IF LEN (a$)<>1 THEN ERROR 100
 40 GOTO 20
100 IF ERR=100 THEN 110 ELSE 130
110 PRINT CHR$ (7)
120 PRINT "powiedziałem JEDEN znak!"
130 RESUME 20
run
```

Słowa kluczowe pokrewne: ERL, ERR, ON ERROR GOTO, RESUME

EVERY

EVERY <okres czasu> [,<numer licznika>] GOSUB<numer linii>

```
10 EVERY 50,1 GOSUB 30
20 GOTO 20
30 SOUND 1,20
40 RETURN
run
```

ROZKAZ: Wywołuje podprogram w BASIC'u w regularnych odstępach czasu, określonych przez <okres czasu> w jednostkach 0,02 sekundowych (EVERY - każdy).

<Numer licznika> (zakres 0 do 3) określa, który spośród czterech liczników ma być użyty. Licznik 3 ma najwyższy priorytet, licznik 0 (przyjmowany w przypadku braku określenia) najniższy.

Każdy licznik może wywoływać swój podprogram.

Dalsze informacje dotyczące przerwań można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: AFTER, REMAIN

EXP

EXP (<wyrażenie liczbowe>)

```
PRINT EXP (6.876)
968.743625
```

FUNKCJA: Oblicza "e" do potęgi podanej w <wyrażeniu liczbowym>, gdzie "e" równe około 2,7182818 jest liczbą, której logarytm naturalny jest równy 1.

Słowa kluczowe pokrewne: LOG

FILL

FILL <atrament>

```
10 MODE 0
20 FOR n=1 TO 500
30 PRINT "O"
40 NEXT
50 kolor pióra =2+RND*13
60 FILL kolor pióra
70 GOTO 50
run
```

ROZKAZ: Wypełnia (FILL) wybrany obszar ekranu graficznego. Krańce obszaru są ograniczone liniami narysowanymi aktualnym atramentem pióra graficznego, lub atramentem użytym do wypełniania (zakres <atramentu> 0 do 15).

Wypełnianie rozpoczyna się od bieżącej pozycji kursora graficznego. Jeżeli pozycja ta leży na linii ograniczającej, wypełnienie nie nastąpi.

Słowa kluczowe pokrewne: GRAPHICS PEN

FIX

FIX (<wyrażenie liczbowe>)

```
PRINT FIX (9.99999)
9
```

FUNKCJA: Usuwa część ułamkową wyrażenia liczbowego zaokrąglając ją do zera (FIX - ustalać).

Słowa kluczowe pokrewne: CINT, INT, ROUND

FN

(Zobacz DEF FN)

FOR

FOR <zmienna jednoliterowa> = <początek> TO <koniec>
[STEP <wartość kroku>]

```
10 FOR n=2 TO 8 STEP 2
20 PRINT n;
30 NEXT n
40 PRINT ",zagrałem ci na nosie"
run
```

ROZKAZ: Wykonuje program zawarty pomiędzy rozkazami FOR (dla) i NEXT określoną liczbą razy, zmieniając zmienną sterującą od < początku > do <końca> o parametr < wartość kroku >. Jeżeli < wartość kroku > nie jest podana, przyjmowane jest 1.

<Wartość kroku> może być ujemna i wówczas wartość <początku> musi być większa niż wartość <końca> ,w przeciwnym wypadku zmienna sterująca nie będzie zmieniana.

Pętle FOR NEXT mogą być "zagnieżdżone" wielokrotnie jedna w drugiej.

Dopisywanie nazwy zmiennej do rozkazu NEXT nie jest obowiązkowe, ponieważ BASIC potrafi znaleźć, który rozkaz FOR jest skojarzony z "anonimowym" rozkazem NEXT.

Słowa kluczowe pokrewne: NEXT, STEP, TO

FRAME

FRAME

```
 10 MODE 0
 20 PRINT "FRAME wyłączone"
 30 TAG
 40 MOVE 0,200
 50 FOR x=0 TO 500 STEP 4
 60 IF f=1 THEN FRAME
 70 MOVE x, 200
 80 PRINT " "; CHR$ (143);
 90 NEXT
100 IF f=1 THEN RUN
110 CLS
120 TAGOFF
130 PRINT "FRAME włączone"
140 f=1
150 GOTO 30
run
```

ROZKAZ: Synchronizuje (FRAME - dopasować) wpisywanie znaków graficznych na ekran w czasie powrotów plamki świetlnej. Ogólnym efektem tego jest gładki ruch elementów po ekranie, bez "migotania" lub "rozdzierania".

Słowa kluczowe pokrewne: TAG, TAGOFF

FRE

FRE (<wyrażenie liczbowe>)
FRE (<wyrażenie tekstowe>)

```
PRINT FRE (0)
PRINT FRE (" ")
```

FUNKCJA: Podaje ile wolnej (FREe) pamięci pozostaje niezajęte przez BASIC. Forma FRE (" ") powoduje "usunięcie śmieci" przed podaniem wartości dostępnego miejsca.
Uwaga. BASIC używa tylko pierwszych 64 K pamięci.
Słowa kluczowe pokrewne: HIMEM, MEMORY

GOSUB

GOSUB <numer linii>

```
GOSUB 210
```

ROZKAZ: Powoduje przejście do podprogramu (GO to a SUBroutine) w BASIC'u, rozpoczynającego się od podanego <numeru linii>. Koniec podprogramu jest zaznaczony rozkazem RETURN, po którym wykonywanie programu jest kontynuowane od instrukcji występującej po wywołującym rozkazie GOSUB.
Słowa kluczowe pokrewne: RETURN

GOTO

GOTO <numer linii>

```
GOTO 90
```

ROZKAZ: Powoduje przejście do (GO TO) podanej linii.
Słowa kluczowe pokrewne: nie ma

GRAPHICS PAPER

GRAPHICS PAPER <atrament>

```
10 MODE 0
20 MASK 15
30 GRAPHICS PAPER 3
40 DRAW 640, 0
run
```

ROZKAZ: Przypisuje <atrament> do papieru graficznego (GRAPHICS PAPER). Przy rysowaniu linii ciągłych papier graficzny nie jest widoczny. W powyższym przykładzie rozkaz MASK umożliwia rysowanie linii przerywanej, co czyni papier graficzny widzialny.

<Atrament> papieru graficznego (w zakresie 0 do 15) jest również używany do tła znaków wpisanych rozkazem TAG oraz przy czyszczeniu okienek graficznych rozkazem CLG (w przypadku braku innego określenia).

Słowa kluczowe pokrewne: CLG, GRAPHICS PEN, INK, MASK, TAG, TAGOFF

GRAPHICS PEN

GRAPHICS PEN [<atrament>] [,<rodzaj tła>]

```
10 MODE 0
20 GRAPHICS PEN 15
30 MOVE 200, 0
40 DRAW, 200, 400
50 MOVE 639, 0
60 FILL 15
run
```

ROZKAZ: Określa, który <atrament> (w zakresie 0 do 15) ma być użyty do rysowania linii i punktów (GRAPHICS PEN - pióro graficzne) <Rodzaj tła> może być:

0: Tło nieprzezroczyste

1: Tło przezroczyste

(Tło przezroczyste powoduje, że na ekranie widoczne są znaki wpisane z rozkazem TAG i przerwy w liniach kropkowanych).

Można pominąć jeden z parametrów rozkazu, ale nie obydwa. Jeżeli któryś z parametrów jest pominięty, jego określenie nie zmienia się.

Słowa kluczowe pokrewne: GRAPHICS PAPER, INK, MASK, TAG, TAGOFF

HEX$

HEX$ (< wyrażenie określające liczbę całkowitą bez znaku> [,<szerokość pola>]

```
PRINT HEX$ (255,4)
00FF
```

FUNKCJA: Daje ciąg cyfr hexadecymalnych (HEXadecimal $tring) reprezentujących wartość < wyrażenia określającego liczbę całkowitą bez znaku > . Długość ciągu wyznacza < szerokość pola > (w zakresie 0 do 16). Jeżeli żądana liczba cyfr jest za duża, wynik jest zapełniany nieznaczącymi zerami; jeżeli żądana liczba cyfr jest za mała, wynik NIE jest skracany do tej długości, lecz jest podawany z taką liczbą cyfr jaka jest potrzebna.

Zakres wartości < wyrażenia określającego liczbę całkowitą bez znaku > jest -32768 do 65535.

Słowa kluczowe pokrewne: BIN$, DEC$, STR$, UNT

HIMEM

HIMEM

```
PRINT HIMEM
42619
```

FUNKCJA: Podaje adres najwyższego (HIghest) bajtu pamięci (MEMory) używanej przez BASIC (który może być zmieniony rozkazem MEMORY).

Uwaga. BASIC używa tylko pierwszych 64 K pamięci.

Słowa kluczowe pokrewne: FRE, MEMORY, SYMBOL, SYMBOL AFTER

IF

IF < wyrażenie logiczne > THEN < opcja > ELSE < opcja >

```
10 MODE 1
20 x=CINT (RND*100)
30 PRINT "Zgadnij mój numer (0 do 100)"
40 INPUT n
50 IF n<x THEN PRINT n; "jest za małe ..."
60 IF n>x THEN PRINT n; "jest za duże ..."
70 IF n=x THEN 80 ELSE c=c+1:GOTO 40
80 PRINT "Doskonale, trafiłeś";
90 PRINT c+1; "trafień!"
run
```

ROZKAZ: Jeżeli (IF) wartość <wyrażenia logicznego> jest "prawda" to (THEN) wykonywana jest pierwsza <opcja>. Jeżeli wartość <wyrażenia logicznego> jest "fałsz" i istnieje druga <opcja>, to jest ona wykonywana; w przeciwnym wypadku wykonywany jest następny rozkaz.

Rozkazy IF THEN mogą być "zagnieżdżane" do dowolnej głębokości i ich działanie kończy się z końcem linii. Dlatego nie można w jednej linii umieścić rozkazów IF THEN i rozkazów od nich niezależnych.

Jeżeli <opcją> jest skok do linii o danym numerze, rozkaz taki może być napisany jak na przykładach:

```
IF a = 1 THEN 100
```

... albo

```
IF a = 1 GOTO 100
```

... albo

```
IF a = 1 THEN GOTO 100
```

Słowa kluczowe pokrewne: ELSE, GOTO, THEN

INK

INK <atrament> ,<kolor> [, <kolor>]

```
10 MODE 1:PAPER 0:PEN 1
20 FOR p=0 TO 1
30 FOR i=0 TO 26
40 INK  p,i
50 LOCATE 16,12:PRINT "atrament"; p; ",";i
60 FOR t=1 TO 400:NEXT t,i,p
70 INK 0,1:INK 1,24:CLS
run
```

ROZKAZ: Przypisuje kolor(y) do danego atramentu (INK). Parametr <atrament> musi być liczbą całkowitą z zakresu 0 do 15 i może być następnie stosowany w rozkazach PEN i PAPER. Pierwszy <kolor> powinien być liczbą z zakresu 0 do 26. Jeżeli podany jest drugi <kolor>, atrament zmienia barwę z szybkością określoną przez rozkaz SPEED INK.

Słowa kluczowe pokrewne: GRAPHICS PAPER, GRAPHICS PEN, PAPER, PEN, SPEED INK

INKEY

INKEY (<wyrażenie określające liczbę całkowitą>)

```
10 IF INKEY (55) <> 32 THEN 10
20 PRINT "Nacisnąłeś [SHIFT] i V"
30 CLEAR INPUT
run
```

FUNKCJA: Podaje stan klawiszów: [SHIFT], [CONTROL] i klawisza (KEY) o danym numerze. Klawiatura jest przeglądana co 0,02 sekundy.

Funkcja ta jest użyteczna w przypadku sprawdzania czy dany klawisz jest wciśnięty. Sprawdza się wówczas czy wartość funkcji jest równa -1 (niezależne od stanu klawiszów [SHIFT] i [CONTROL]).

Powyższy przykład sprawdza czy wciśnięty jest [SHIFT] i V (klawisz nr. 55). Wykaz numerów klawiszów umieszczono na pokrywie komputera oraz w rozdziale zatytułowanym "Nieco użytecznych informacji ...".

Stan klawiszów określony jest w sposób następujący:

Wartość funkcji	[SHIFT]	[CONTROL]	dany klawisz
-1	nie zależy	nie zależy	zwolniony
0	zwolniony	zwolniony	wciśnięty
32	wciśnięty	zwolniony	wciśnięty
128	zwolniony	wciśnięty	wciśnięty
160	wciśnięty	wciśnięty	wciśnięty

Słowa kluczowe pokrewne: CLEAR INPUT, INKEY$, JOY

INKEY$

INKEY$

```
10 CLS
20 PRINT "Wybierz Tak lub Nie (T/N)?"
30 a$=INKEY$
40 IF a$="" THEN 30
50 IF a$="t" OR a$="T" THEN 80
60 IF a$="n" OR a$="N" THEN 90
70 GOTO 30
80 PRINT "Wybrałeś TAK":END
90 PRINT "Wybrałeś NIE"
run
```

FUNKCJA: Podaje ciąg ($tring) znaków (znak) z klawiatury (KEYboard). Jeżeli żaden klawisz nie jest wciśnięty, ciąg jest pusty. W powyższym przykładzie sprawdza się czy wciśnięty jest klawisz "Y", czy "N", czy żaden z nich.
Słowa kluczowe pokrewne: CLEAR INPUT, INKEY

INP

INP (<numer portu>)

```
PRINT INP (&FF77)
```

FUNKCJA: Podaje wartość z portu wejściowego (INput Port) o adresie określonym przez <numer portu>.
Słowa kluczowe pokrewne: OUT, WAIT

INPUT

INPUT [#<wyrażenie określające strumień>,] [;] [<tekst do powtórzenia> <separator>] <lista: <zmiennych>

```
10 MODE 1
20 INPUT "Podaj dwie liczby do mnożenia (oddzielone
   przecinkiem)"; a,b
30 PRINT a; "razy"; b; "równa się"; a*b
40 GOTO 20
run
```

ROZKAZ: Przyjmuje dane z danego strumienia (ze strumienia #0 w przypadku braku określenia) (INPUT -wejście).

Pierwszy, opcjonalny średnik [;] likwiduje przesunięcie kursora na początek nowej linii po wprowadzeniu danej.

<Separator> musi być średnikiem lub przecinkiem. Średnik powoduje, że wyświetlany jest pytajnik, przecinek powoduje, że pytajnik nie jest wyświetlany.

Jeżeli na wejściu pojawia się dana złego typu (na przykład jeżeli wpisana jest przez pomyłkę litera O zamiast 0 - zero, przy przyjmowaniu danych liczbowych), BASIC wyświetla

```
? Redo from start
```

...i tekst do powtórzenia .
Wpisywanie każdej danej musi być zakończone klawiszem RETURN
Słowa kluczowe pokrewne: LINE INPUT

INSTR

INSTR ([<pozycja startowa>,] <tekst przeszukiwany> , <tekst poszukiwany>)

```
10 CLS:FOR n=1 TO 26
20 alfabet$=alfabet$+CHR$(n+64)
30 NEXT
40 INPUT "Wpisz literę"; a$
50 b$=UPPER$(a$)
60 PRINT b$;"jest"
70 PRINT INSTR (alfabet$,b$)
80 PRINT "literą w alfabecie.":PRINT
90 GOTO 40
run
```

FUNKCJA: Przeszukuje <tekst przeszukiwany> w celu znalezienia <tekstu poszukiwanego> i podaje pozycję w <tekście przeszukiwanym> pierwszego znalezionego <tekstu poszukiwanego>. Jeżeli poszukiwanie jest bezskuteczne, wynik jest 0.

Pozycja, od której należy rozpocząć przeszukiwanie może być zadana parametrem <pozycją startowa>, którego wartość musi być liczbą całkowitą od 1 do 255.

Słowa kluczowe pokrewne: nie ma.

INT

INT (<wyrażenie liczbowe>)

```
PRINT INT (-1.995)
-2
```

FUNKCJA: Zaokrągla liczbę do najbliższej, mniejszej liczby całkowitej (INTeger). Podaje te same wartości co FIX, dla liczb dodatnich, ale o jeden mniej niż FIX dla liczb ujemnych, które nie są liczbami całkowitymi.

Słowa kluczowe pokrewne: CINT, FIX, ROUND

JOY

JOY (<wyrażenie określające liczbę całkowitą>)

```
10 PRINT "Aby zatrzymać program-";
20 PRINT "użyj joysticka"
30 IF JOY (0) <> 0 THEN END
40 GOTO 10
run
```

FUNKCJA: Odczytuje znaczoną poszczególnymi bitami pozycję JOYstick'a, którego numer (0 lub 1) podano w <wyrażeniu określającym liczbę całkowitą>.

Bit	Dziesiętnie
0: W górę	1
1: W dół	2
2: W lewo	4
3: W prawo	8
4: Spust 2	16
5: Spust 1	32

Stąd na przykład, jeżeli główny spust (spust 2) pierwszego joystick'a jest naciśnięty, kiedy dźwignia joystick'a jest w pozycji w lewo, funkcja JOY (0) daje wartość dziesiętną 20 (odpowiednio 16+4).

Dalsze informacje dotyczące joystick'ów można znaleźć w rozdziale zatytułowanym "Nieco użytecznych informacji ..."

Słowa kluczowe pokrewne: CLEAR INPUT, INKEY

KEY

KEY <numer ekspansji znakowej>, <wyrażenie tekstowe>

```
KEY 11,"border 13:paper 0:pen 1:ink 0,13:
ink 1,0:mode 2:list"+CHR$ (13)
```

...teraz wciśnij klawisz ENTER

ROZKAZ: Przypisuje <wyrażenie tekstowe> do klawisza (KEY), którego <numer ekspansji znakowej> podano. Możliwe są trzydzieści dwa"znaki rozszerzone", o numerach 0 do 31, przypisane do wartości kodu klawiszów 128 do 159. Klawisze o wartościach 128 (*f*0 w bloku klawiszów cyfrowych) do 140 ([CONTROL][ENTER]) są, w braku określenia, przypisane do cyfr 0 do 9, kropki dziesiętnej, kodu RETURN i RUN " RETURN (dla operacji z kasetą magnetofonową), ale mogą być wtórnie przypisane do innych <wyrażeń tekstowych>. "Znaki rozszerzone" 13 do 31 (wartości klawiszów 141 do 159) przypisane są, w braku określenia do tekstów pustych, ale mogą być wypełnione i przypisane do klawiszów przy użyciu rozkazu KEY DEF opisanego w następnym przykładzie.

<Numer ekspansji znakowej> może być w zakresie 0 do 31 lub, opcjonalnie, 128 do 159 dla odzwierciedlenia wartości klawiszów. (Zobacz ilustrację klawiatury w rozdziale zatytułowanym "Nieco użytecznych informacji ..."

<Wyrażenie tekstowe> może mieć długość 120 znaków. W przypadku przekroczenia tej wartości wyświetlany jest błąd nr 5 - Improper argument

Słowa kluczowe pokrewne: KEY DEF

KEY DEF

KEY DEF <numer klawisza>, <powtarzanie> [,<wartość normalna> [,<z klawiszem [SHIFT]> [,<z klawiszem [CONTROL]]]

```
KEY 159 "to jest klawisz tabulacji"
KEY DEF 68,1,159
```

... teraz wciśnij klawisz TAB

ROZKAZ: Określa (DEFine) wartość klawisza (KEY) o danym <numerze klawisza> w zakresie 0 do 79 (numery klawiszów podane są na pokrywie komputera i w rozdziale zatytułowanym "Nieco użytecznych informacji ..."). <Wartość normalna>, <z klawiszem [SHIFT]> i <z klawiszem [CONTROL]> powinny określać wartości kodów jakie ma dawać wciśnięcie danego klawisza odpowiednio: samego, razem z klawiszem SHIFT i razem z klawiszem CONTROL. Parametry te są opcjonalne.

Parametr <powtarzanie> pozwala włączyć lub wyłączyć (1 lub 0) automatyczne powtarzanie kodu klawisza. Częstotliwość <powtarzania> może być zmieniana rozkazem SPEED KEY.

W powyższym przykładzie klawisz o wartości 159 (odpowiadającej numerowi znaku rozszerzonego 31) przypisany jest najpierw "znak rozszerzony": "to jest klawisz tabulacji", następnie rozkaz KEY DEF określa klawisz 68 (TAB) jako powtarzający (1) o kodzie 159.

W powyższym przykładzie normalne działanie może być przywrócone przez:

```
KEY DEF 68,0,9
```

... gdzie 9 jest normalną wartością klawisza TAB

Słowa kluczowe pokrewne: KEY, SPEED KEY

LEFT$

LEFT$ (<wyrażenie tekstowe>, <żądana długość>)

```
10 CLS
20 a$="AMSTRAD"
30 FOR n=1 TO 7
40 PRINT LEFT$ (a$,n)
50 NEXT
run
```

FUNKCJA: Podaje ciąg znaków, o długości (w zakresie 0 do 255) określonej przez <żądaną długość>, wyodrębniony z lewej (LEFT) strony <wyrażenia tekstowego>. Jeżeli <wyrażenie tekstowe> jest krótsze niż <żądana długość> podawane jest całe <wyrażenie tekstowe>.
Słowa kluczowe pokrewne: MID$, RIGHT$

LEN

LEN (<wyrażenie tekstowe>)

```
10 LINE INPUT "Wpisz jakieś zdanie"; a$
20 PRINT "Zdanie ma";
30 PRINT LEN (a$); "znaków."
run
```

FUNKCJA: Podaje całkowitą długość (LENgth) <wyrażenia tekstowego>.
Słowo kluczowe pokrewne: nie ma

LET

LET <zmienna> = <wyrażenie>

```
LET x=100
```

ROZKAZ: Przypisuje wartość do zmiennej. Pozostałość po wczesnych BASIC'ach, w których podstawianie do zmiennych musiało być tak oznaczone (LET - niech). Nie ma zastosowania w AMSTRAD BASIC prócz zapewnienia kompatybilności z programami podanymi w starych podręcznikach BASIC'a. Powyższy przykład powinien być napisany po prostu:

```
x=100
```

Słowa kluczowe pokrewne: nie ma

LINE INPUT

LINE INPUT [#<wyrażenie określające strumień>,] [;] [<tekst do powtórzenia><separator>]<zmienna tekstowa>

```
10 LINE INPUT "Wpisz jedną linię tekstu",a$
20 CLS
30 PRINT "Zmienna a$ jest teraz równa:-"
40 PRINT a$
run
```

ROZKAZ: przyjmuje całą linię tekstu ze wskazanego strumienia (strumień #0 w braku określenia). Pierwszy, opcjonalny średnik powoduje, że kursor nie jest przestawiany na początek nowej linii po wykonaniu rozkazu.

<Separator> musi być średnikiem lub przecinkiem. Średnik powoduje wyświetlenie pytajnika, przecinek przeciwnie.

LINE INPUT z klawiatury (#0) jest kończony przez naciśnięcie [RETURN].

LINE INPUT z dysku (lub kasety), strumień #9 jest kończony znakiem CR (0DH w kodzie ASCII) lub w chwili przyjęcia 255-tego znaku.

Słowa kluczowe pokrewne: INPUT

LIST

LIST [<zakres numerów linii>] [,#<wyrażenie określające strumień>]

```
LIST 100-1000,#1
```

ROZKAZ: LISTuje linie programu w danym strumieniu. Strumień #0 (przyjmowany w braku określenia) jest strumieniem ekranu, #8 strumieniem drukarki. Listowanie można przerwać przez jednokrotne naciśnięcie [ESC] i ponownie uruchomione przez naciśnięcie spacji. Dwukrotne naciśnięcie [ESC] powoduje zaprzestanie listowania i powrót do trybu bezpośredniego.

Pierwszy lub ostatni numer w <zakresie numerów linii> może być pominięty dla wskazania: "... od początku programu" lub "... do końca programu"; na przykład:

```
LIST -200
```

... lub ...

```
LIST 50-
```

... lub ...

```
LIST
```

co listuje cały program.

Słowa kluczowe pokrewne: nie ma

LOAD

LOAD <nazwa zbioru> [,<wyrażenie określające adres>]

```
LOAD "discfile.xyz",&2AF8
```

ROZKAZ: Ładuje (LOAD) program BASIC'owy z dysku do pamięci, zastępując program znajdujący się w pamięci. Podanie opcjonalnego <wyrażenia określającego adres> powoduje ładowanie zbioru binarnego pod wskazany adres, niezależnie od adresu, z którego zbiór ten był zapisywany.

Zabezpieczone programy BASIC'owe nie mogą być ładowane tym rozkazem, ponieważ są one wówczas natychmiast usuwane z pamięci. Zamiast tego rozkazu należy używać wtedy rozkazów RUN lub CHAIN.

Słowa kluczowe pokrewne: CHAIN, CHAIN MERGE, MERGE, RUN, SAVE

LOCATE

LOCATE [#<wyrażenie określające strumień>,] <współrzędna x>, <współrzędna y>

```
10 MODE 1
20 FOR n=1 TO 20
30 LOCATE n,n
40 PRINT CHR$(143);"location";
50 PRINT n;",";n
60 NEXT
run
```

ROZKAZ: Umieszcza (LOCATE) kursor tekstowy, we wskazanym strumieniu, na pozycji określonej przez współrzędne x i y. Pozycja 1,1 znajduje się w lewym górnym rogu strumienia (okna). Strumień #0 przyjmowany jest w braku określenia.

Słowa kluczowe pokrewne: WINDOW

LOG

LOG (<wyrażenie liczbowe>)

```
PRINT LOG (9999)
9.21024037
```

FUNKCJA: Oblicza LOGarytm naturalny <wyrażenia liczbowego>, które musi być większe od zera.
Słowa kluczowe pokrewne: EXP, LOG 10

LOG 10

LOG 10 (<wyrażenie liczbowe>)

```
PRINT LOG 10 (9999)
3.99995657
```

FUNKCJA: Oblicza LOGarytm dziesiętny <wyrażenia liczbowego>, które musi być większe od zera.
Słowa kluczowe pokrewne: EXP, LOG

LOWER$

LOWER$ (<wyrażenie tekstowe>)

```
10 a$ = "POPATRZ JAK LITERY ZMIENIAJĄ SIĘ"
20 PRINT LOWER$ (a$+"NA MALE")
run
```

FUNKCJA: Podaje nowy tekst, który jest kopią tekstu z <wyrażenia tekstowego>, ale w którym wszystkie duże litery zamienione są na małe (LOWER) litery. Funkcja ta jest użyteczna do przetwarzania ciągów wejściowych, które mogą zawierać pomieszane małe i duże litery.
Słowa kluczowe pokrewne: UPPER$

MASK

MASK [<wyrażenie określające liczbę całkowitą>][,<ustawienie pierwszego punktu>]

```
10 MODE 0:INK 5,21:INK 8,16
20 MOVE -100*RND, 400*RND
30 WHILE XPOS < 640
40 FOR x=1 TO 8
50 MASK 2↑(8-x)
60 DRAWR 32,0,x,1:MOVER -32,0
70 NEXT
80 MOVER 34,0
90 WEND:GOTO 20
run
```

ROZKAZ: Określa "MASKę" lub szablon, który ma być użyty przy rysowaniu linii. Wartość binarna <wyrażenia określającego liczbę całkowitą>, w zakresie 0 do 255, włącza (1) lub wyłącza (0) bity w każdej grupie 8 kolejnych punktów ekranu.

Ustawienie pierwszego punktu decyduje o tym, czy pierwszy punkt linii ma być rysowany (1) czy nie (0).

Każdy z parametrów może być pominięty, ale nie obydwa na raz. Jeżeli parametr jest pominięty, nie zmienia się właściwość, którą on określa.

Słowa kluczowe pokrewne: DRAW, DRAWR, GRAPHICS PAPER, GRAPHICS PEN

MAX

MAX (<lista:<wyrażeń liczbowych>)

```
10 n=66
20 PRINT MAX (1,n,3,6,4,3)
run
66
```

FUNKCJA: Podaje MAXymalną wartość <wyrażeń liczbowych> umieszczonych w liście.

Słowa kluczowe pokrewne: MIN

MEMORY

MEMORY wyrażenie określające adres

```
MEMORY &20AA
```

ROZKAZ: Zmienia ilość pamięci (MEMORY) dostępnej dla BASIC'u, poprzez określenie adresu najwyżej położonego bajtu.

Uwaga. BASIC używa tylko pierwszych 64 K pamięci.

Słowa kluczowe pokrewne: FRE, HIMEM, SYMBOL, SYMBOL AFTER

MERGE

MERGE <nazwa zbioru>

ROZKAZ: ładuje program z dysku i łączy (MERGE) go z programem zawartym w pamięci.

Uwaga, linie programu z dysku o tych samych numerach co linie starego programu zastępują te ostatnie.

Zbiory zabezpieczone (z przyrostkiem ,p) nie mogą być dołączane do programu tym rozkazem.
Słowa kluczowe pokrewne: CHAIN, CHAIN MERGE, LOAD

MID$

MID$ (<wyrażenie tekstowe> , <pozycja startowa> [,<żądana długość>)

```
 10 MODE 1:ZONE 3
 20 a$="ENCYKLOPEDIA"
 30 PRINT "Przeliteruj ";a$
 40 PRINT "OK....":PRINT
 50 FOR n=1 TO LEN (a$)
 60 PRINT MID$ (a$,n,1)
 70 FOR t=1 TO 700:NEXT t,n
 80 PRINT:PRINT
 90 INPUT "Teraz wpisz inne słowo"; a$
100 GOTO 50
run
```

FUNKCJA: Podaje nowy tekst, o żądanej długości , będący wyciągiem z tekstu określonego <wyrażeniem tekstowym>, rozpoczynającym się od danej <pozycji startowej>. Jeżeli nie jest podana <żądana długość>, podawana jest cała reszta tekstu od <pozycji startowej>do końca.

Jeżeli <pozycja startowa> jest większa niż długość tekstu <wyrażenia tekstowego> , ciąg wynikowy jest ciągiem pustym. Zakres <pozycji startowej> jest 0 do 255, zakres żądanej długości 0 do 255.
Słowa kluczowe pokrewne: LEFT$, RIGHT$

MID$

MID$ (<zmienna tekstowa>,<pozycja wtrącenia> [,<długość nowego tekstu>) = <nowe wyrażenie tekstowe>

```
10 a$= hello"
20 MIDS(a$,3,2)="XX"
30 PRINT a$
run
heXXo
```

ROZKAZ: Wprowadza <nowe wyrażenie tekstowe> w tekst przechowywany w <zmiennej tekstowej>, rozpoczynając od <pozycji wtrącenia> i zajmując liczbę znaków tekstu równą <długości nowego tekstu>.

Uwaga, przy użyciu MID$ jako rozkazu, tekst pierwotny musi być podany w postaci zmiennej tekstowej (np. a$) nie może to być stała tekstowa (np. "hello").

Słowa kluczowe pokrewne: LEFT$, RIGHT$

MIN

MIN (<lista: <wyrażeń liczbowych>)

```
PRINT MIN (3,6,2.999,8,9)
2.999
```

FUNKCJA: Podaje MINimalną wartość spośród <wyrażeń liczbowych> występujących w liście.

Słowa kluczowe pokrewne: MAX

MOD

<argument> MOD <argument>

```
PRINT 10 MOD 3
1
PRINT 10 MOD 5
0
```

OPERATOR: Wynikiem jest reszta z dzielenia pierwszego <argumentu> przez drugi (dzielenie MODulo X).

Słowa kluczowe pokrewne: nie ma

MODE

MODE <wyrażenie określające liczbę całkowitą>

```
10 m=m+1:IF m>2 THEN m=0
20 MODE m
30 PRINT "to jest tryb"; m
40 PRINT "teraz naciśnij jakiś klawisz"
50 IF INKEY$="" THEN 50 ELSE 10
run
```

ROZKAZ: Zmienia tryb (MODE) organizacji ekranu (zakres 0 do 2) i czyści ekran kolorem atramentu 0 (,który nie musi być aktualnym atramentem papieru). Wszystkie okna tekstowe i graficzne są likwidowane, kursor tekstu i graficzny są przesuwane do ich pozycji wyjściowych.
Słowa kluczowe pokrewne: ORIGIN, WINDOW

MOVE

MOVE <współrzędna x>, <współrzędna y> [,[<atrament>]
[,<tryb rysowania>]]

```
10 MODE 1:TAG
20 x=RND*800-100:y=RND*430
30 MOVE x,y
40 PRINT "jestem tutaj";
50 GOTO 20
run
```

ROZKAZ: Przesuwa (MOVE) kursor graficzny do pozycji (x,y). Opcjonalny parametr <atrament> zmienia atrament (w zakresie 0 do 15) przypisany do pióra graficznego.

Opcjonalny parametr <tryb rysowania> określa rodzaj interakcji atramentu z liniami narysowanymi wcześniej.
Tryby rysowania są następujące:
0: Normalny
1: XOR (eXclusive OR)
2: AND (i)
3: OR (lub)
Słowa kluczowe pokrewne: MOVER, ORIGIN, XPOS, YPOS

MOVER

MOVER <wektor x>, <wektor y> [, [<atrament>]
[,<tryb rysowania>]]

```
10 MODE 1:TAG:MOVE 0,16
20 PRINT "życie ma swoje"
30 FOR n=1 TO 10
40 MOVER -32,16
50 PRINT "wzloty";:NEXT:PRINT "i";
60 FOR n=1 TO 10
70 MOVER -64, -16
80 PRINT "upadki";:NEXT
run
```

ROZKAZ: Przesuwa względnie (MOVE Relative) kursor graficzny od bieżącej pozycji o wskazany wektor. Opcjonalny parametr <atrament> zmienia atrament (w zakresie 0 do 15) przypisany do pióra graficznego.

Opcjonalny parametr <tryb rysowania> określa rodzaj interakcji atramentu z liniami narysowanymi wcześniej. Tryby rysowania są następujące:

0: Normalny
1: XOR (EXclusive OR)
2: AND (i)
3: OR (lub)

Słowa kluczowe pokrewne: MOVE, ORIGIN, XPOS, YPOS

NEW

NEW

```
NEW
```

ROZKAZ: Usuwa program i zeruje zmienne w pamięci. Nie usuwa definicji klawiszów, nie zmienia charakterystyk ekranu (MODE, PEN, INK itd.), nie czyści ekranu.

Słowa kluczowe pokrewne: nie ma.

NEXT

NEXT <lista:<zmiennych>

```
10 FOR a=1 TO 3
20 FOR b=0 TO 26
30 MODE 1
40 PEN a:BORDER b
50 PRINT "pen"; a; "border"; b
60 FOR c=1 TO 500
70 NEXT c,b,a
run
```

ROZKAZ: Zamyka pętlę FOR. Rozkaz ten może być anonimowy, lub może odnosić się do odpowiedniego FOR. Jak w powyższym przykładzie, <lista:<zmiennych> musi mieć odwrotną kolejność w porównaniu z występowaniem rozkazów FOR, tak aby pętle "zagnieżdżone" nie przecinały się.

Słowa kluczowe pokrewne: FOR, STEP, TO

NOT

NOT argument

```
IF NOT "piotr"<"zenek" THEN PRINT "dobrze" ELSE PRINT "źle"
źle
IF NOT "kot" >"pies"  THEN PRINT "dobrze" ELSE PRINT "źle"
dobrze
...
PRINT NOT -1
0
PRINT NOT 0
-1
```

OPERATOR: Wykonuje operację logiczną na bitach liczb całkowitych. Neguje każdy bit argumentu.

Dalsze informacje dotyczące logiki mogą być znalezione w 2. części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: AND, OR, XOR

ON BREAK CONT

ON BREAK CONT

```
10 ON BREAK CONT
20 PRINT "Program będzie kontynuowany nawet jeżeli spróbu-
   jesz go przerwać klawiszem  ESC ": PRINT
30 FOR t=1 TO 1000:NEXT:GOTO 20
run
```

ROZKAZ: Powoduje kontynuowanie wykonywania programu, pomimo przerwania klawiszem [ESC]. Należy być ostrożnym przy używaniu tego rozkazu, ponieważ przerwać wykonywanie programu można wówczas wyłącznie przez wyzerowanie komputera, stąd program taki powinien być przed uruchomieniem zapisany na dysku.

Działanie tego rozkazu może być zniesione w programie rozkazem ON BREAK STOP.

Słowa kluczowe pokrewne: ON BREAK GOSUB, ON BREAK STOP

ON BREAK GOSUB

ON BREAK GOSUB <numer linii>

```
10 ON BREAK GOSUB 40
20 PRINT "program jest wykonywany"
30 GOTO 20
40 CLS:PRINT "Dwukrotne naciśnięcie [ESC]";
50 PRINT "wywołuje podprogram"
60 FOR t=1 TO 2000:NEXT
70 RETURN
run
```

ROZKAZ: Powoduje skok do podprogramu rozpoczynającego się w linii o podanym numerze, w przypadku dwukrotnego naciśnięcia klawisza [ESC].

Słowa kluczowe pokrewne: ON BREAK CONT, ON BREAK STOP, RETURN

ON BREAK STOP

ON BREAK STOP

```
10 ON BREAK GOSUB 40
20 PRINT "program jest wykonywany"
30 GOTO 20
40 CLS:PRINT "Dwukrotne naciśnięcie  ESC "
50 PRINT "wywołuje podprogram"
60 FOR t=1 TO 2000:NEXT
70 ON BREAK STOP:RETURN
run
```

ROZKAZ: Znosi działanie rozkazów ON BREAK CONT i ON BREAK GOSUB, tak, że klawisz [ESC] odzyskuje możliwość przerywania programu. W powyższym przykładzie rozkaz ON BREAK GOSUB zadziała tylko raz, ponieważ jest następnie zniesiony przez linię 65 podprogramu przerwania.

Słowa kluczowe pokrewne: ON BREAK CONT, ON BREAK GOSUB

ON ERROR GOTO

ON ERROR GOTO <numer linii>

```
10 ON ERROR GOTO 60
20 CLS:PRINT "Jeżeli znajdziesz błąd, ";
30 PRINT "wylistuj program"
40 FOR t=1 TO 4000:NEXT
50 GOTO 100
60 PRINT "Znaleziono błąd w linii";
70 PRINT ERL:PRINT:LIST
run
```

ROZKAZ: Powoduje skok do linii o podanym numerze w przypadku wykrycia błędu w programie.

Rozkaz ON ERROR GOTO Ø przywraca normalną detekcję błędów w BASIC'u.

Zobacz również rozkaz RESUME

Słowa kluczowe pokrewne: DERR, ERL, ERR, ERROR, RESUME

ON <wyrażenie> GOSUB

ON <selektor> GOSUB <lista:< numerów linii>

```
 10 PAPER 0:PEN 1:INK 0,1
 20 CLS:PRINT "WYKAZ OPCJI":PRINT
 30 PRINT "1-Zmiana obrzeża":PRINT
 40 PRINT "2-Zmiana pióra":PRINT
 50 PRINT "3-Zmiana trybu":PRINT
 60 INPUT "Dokonaj wyboru";x
 70 ON x GOSUB 90,110,130
 80 GOTO 20
 90 b=b-1:IF b<0 THEN b=26
100 BORDER b: RETURN
110 p=p-1:IF p<2 THEN p=26
120 INK 1,p:RETURN
130 m=m-1:IF m<0 THEN m=2
140 MODE m:RETURN
run
```

ROZKAZ: Wybiera podprogram, którego numer początkowej linii zależy od wartości <selektora>. Selektor powinien być wyrażeniem dającym liczbę całkowitą w zakresie 0 do 255. Wartość <selektora> określa, który kolejny <numer linii> z listy jest miejscem skoku. W powyższym przykładzie wybór 1 powoduje skok do linii 90, wybór 2 - skok do linii 110, a 3 - do linii 130.

Jeżeli wartość <selektora> jest równa zero, lub jest większa od liczby pozycji w liście, nie jest wybierany żaden podprogram.

Słowa kluczowe pokrewne: RETURN

ON <wyrażenie> GOTO

ON <selektor> GOTO <lista:<numerów linii>

```
 10 CLS:PRINT "WYKAZ OPCJI":PRINT
 20 PRINT "1-Listowanie programu":PRINT
 30 PRINT "2-Edycja":PRINT
 40 PRINT "3-Katalog dysku":PRINT
 50 INPUT "Dokonaj wyboru";n
 60 ON n GOTO 80,90,100
 70 GOTO 10
 80 LIST
 90 AUTO
100 CAT
run
```

ROZKAZ: Wybiera numer linii, do której następuje skok, w sposób zależny od <selektora>. <Selektor> powinien być wyrażeniem dającym liczbę całkowitą w zakresie 0 do 255.

Wartość <selektora> określa, który kolejny <numer linii> z listy jest miejscem skoku. W powyższym przykładzie wybór 1 powoduje skok do linii 80, wybór 2 - skok do linii 90, a 3 do linii 100.

Jeżeli wartość <selektora> jest równa zero, lub jest większa od liczby pozycji w liście, nie jest wykonywany żaden skok.

Słowa kluczowe pokrewne: nie ma

ON SQ GOSUB

ON SQ (<kanał>) GOSUB numer linii

```
10 ENV 1,15,-1,1
20 ON SQ(1) GOSUB 60
30 MODE ":ORIGIN 0,0,200,440,100,300
40 FOR x=1 TO 13:FRAME:MOVE 330,200,x
50 FILL x:NEXT:GOTO 40
60 READ s:IF s=0 THEN RESTORE:GOTO 60
70 SOUND 1,s,25,15,1
80 ON SQ(1) GOSUB 60:RETURN
90 DATA 50,60,90,100,35,200,24,500,0
run
```

ROZKAZ: Wywołuje podprogram w BASIC'u wtedy, kiedy jest wolne miejsce w danej kolejce (Queue) dźwięków (Sound).

Wyrażenie <kanał> powinno mieć jedną z wartości:

1: dla kanału A

2: dla kanału B

4: dla kanału C

Dalsze informacje dotyczące dźwięku można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili ...".

Słowa kluczowe pokrewne: RETURN, SOUND, SQ

OPENIN

OPENIN <nazwa zbioru>

```
10 REM Otwarcie zbioru wejściowego z dysku
20 OPENIN "datafile":INPUT #9,a,a$
30 CLOSEIN:PRINT "2 wartości są następujące:"
40 PRINT:PRINT a,a$
run
```

ROZKAZ: Otwiera (OPEN) zbiór wejściowy (INput) z dysku, do użycia go w bieżącym programie. Zbiór wejściowy musi być zbiorem ASCII.

Powyższy przykład działa tylko wówczas, jeżeli na dysku został zapisany zbiór z następnego przykładu (pod słowem kluczowym OPENOUT).

Słowa kluczowe pokrewne: CLOSEIN,EOF

OPENOUT

OPENOUT <nazwa zbioru>

```
10 REM Otwarcie zbioru wyjściowego na dysk
20 INPUT "podaj wartość zmiennej liczbowej";a
30 INPUT "podaj wartość zmiennej tekstowej";a$
40 OPENOUT "datafile"
50 WRITE #9,a,a$
60 CLOSEOUT:PRINT "Dane      zostały zapisane na dysku"
run
```

ROZKAZ: Otwiera (OPEN) zbiór wyjściowy (OUTput) na dysk.

Słowa kluczowe pokrewne: CLOSEOUT

OR

<argument> OR <argument>

```
IF "piotr"<"zenek" OR "pies" >"kot" THEN PRINT "dobrze"
ELSE PRINT "źle"
dobrze
IF "zenek"<"piotr" OR "kot">"pies" THEN PRINT "dobrze"
ELSE PRINT "źle"
źle
IF "piotr"<"zenek" OR "kot">"pies" THEN PRINT "dobrze"
ELSE PRINT "źle"
dobrze
...
PRINT 1 OR 1
1
PRINT 0 OR 1
1
PRINT 1 OR 0
1
```

OPERATOR: Wykonuje operacje logiczne (OR - lub) na bitach liczb całkowitych. Dany bit wyniku jest równy 1 z wyjątkiem przypadku gdy dane bity argumentów są równe 0.

Dalsze informacje dotyczące logiki można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: AND, NOT, XOR

ORIGIN

ORIGIN <x>,<y>[,<lewa>,<prawa>,<góra>,<dół>]

```
10 MODE 1:BORDER 13:TAG
20 ORIGIN 0,0,100,540,300,100
30 GRAPHICS PAPER 3 CLG
40 FOR x=550 TO -310 STEP -10
50 MOVE x,206
60 PRINT "To jest okno graficzne";
70 FRAME:NEXT:GOTO 10
run
```

ROZKAZ: Ustawia początek (ORIGIN) współrzędnych ekranu graficznego w pozycji określonej przez <x> i <y>.

Przez podanie ostatnich czterech opcjonalnych parametrów określa się wymiary okna graficznego. Jeżeli współrzędne określające okno graficzne opisują punkty leżące poza ekranem, koniec ekranu przyjmowany jest za koniec okna graficznego.

Słowa kluczowe pokrewne: CLG

OUT

OUT <numer portu>,<wyrażenie określające liczbę całkowitą>

```
OUT &8F8F4,&FF
```

ROZKAZ: Wysyła wartość <wyrażenia określającego liczbę całkowitą> (w zakresie 0 do 255) do urządzenia zewnętrznego (OUT) o adresie podanym w <numerze portu>.

Rozkaz ten nie powinien być używany przez niedoświadczonego programistę.

Słowa kluczowe pokrewne: INP, WAIT

PAPER

PAPER [#<wyrażenie określające strumień>,] <atrament>

```
10 MODE 0:PEN 0:INK 0,13
20 FOR p=1 TO 15
30 PAPER p:CLS
40 LOCATE 7,12:PRINT "papier";p
50 FOR t=1 TO 500:NEXT t,p
run
```

ROZKAZ: Ustala atrament tła znaków. Jeżeli znaki są wpisywane w ekran tekstowy, pole znaku jest wypełniane atramentem papieru (PAPER), zanim wpisany jest znak (chyba, że wybrane jest tło przezroczyste).

Jeżeli pominięte jest wyrażenie określające strumień przyjmowany jest strumień # 0.

Liczba różnych atramentów papieru zależy od trybu organizacji ekranu (MODE).

Słowa kluczowe pokrewne: GRAPHICS PAPER, INK, PEN

PEEK

PEEK (<wyrażenie określające adres>)

```
10 MODE 1:ZONE 7
20 WINDOW 1,40,1,2:WINDOW #1,1,40,3,25
30 PRINT "adres pamięci"
40 LOCATE 20,1:PRINT "zawartość pamięci"
50 FOR n=0 TO 65535
60 p=PEEK (n)
70 PRINT #1,n,"(&";HEX$(n);")";
80 PRINT # 1,TAB (20);p,"(&";HEX$(p);")"
90 NEXT
run
```

FUNKCJA: Podaje zawartość komórki pamięci o adresie określonym przez <wyrażenie określające adres> , które powinno być w zakresie &0000 do &FFFF (0 do 65535). W każdym przypadku PEEK podaje wartość odpowiadającą zawartości komórki pamięci RAM (nie ROM) i wartość ta jest w zakresie &00 do &FF (0 do 255)

Słowa kluczowe pokrewne: POKE

PEN

PEN [#<wyrażenie określające strumień>,] [<atrament>][,<rodzaj tła>]

```
10 MODE 0:PAPER 0:INK 0,13
20 FOR p=1 TO 15
30 PEN p:PRINT SPACE$ (47); "pióro";p
40 FOR t=1 TO 500:NEXT t,p:GOTO 20
run
```

ROZKAZ: Określa <atrament> (w zakresie 0 do 15), którym są pisane znaki w danym strumieniu (strumień #0 w braku określenia). Parametr <rodzaj tła> może mieć wartość 1 (tło przezroczyste) lub 0 (tło nieprzezroczyste).

Każdy z dwóch ostatnich parametrów może być pominięty, ale nie obydwa na raz. Jeżeli parametr jest pominięty, właściwość, którą określa, nie zmienia się.

Słowa kluczowe pokrewne: PAPER

PI

PI

```
PRINT PI
3.14259265
```

FUNKCJA: Podaje wartość stosunku obwodu okręgu do jego średnicy.

Słowa kluczowe pokrewne: DEG, RAD

PLOT

PLOT <współrzędna x>,<współrzędna y> [, <atrament>] [<,tryb rysowania>]]

```
10 MODE 1:BORDER 0:PAPER 0:PEN 1
20 INK 0,0:INK 1,26:INK 2,13,26:DEG
30 FOR x=1 TO 360:ORIGIN 320,200
40 DRAW 50*COS(x),50*SIN(x),1
50 PLOT 100*COS(x),25*SIN(x):NEXT
60 ORIGIN 0,0: t=TIME+700:WHILE TIME<t
70 PLOT RND*640,RND*400:WEND
80 PLOT RND*640, RND*400,2
90 GOTO 90
run
```

ROZKAZ: Rysuje punkt na ekranie graficznym na pozycji określonej wartościami współrzędnych. W rozkazie tym może być podany <atrament>, którym punkt ma być rysowany (zakres 0 do 15).

Opcjonalny parametr <tryb rysowania> określa rodzaj interakcji atramentu z punktem już narysowanym na ekranie graficznym. Są 4 tryby rysowania :

0: Normalny

1: XOR (eXclusive OR)

2: AND (i)

3: OR (lub)

Słowa kluczowe pokrewne: GRAPHICS PEN, PLOTR

PLOTR

PLOTR <wektor x> ,<wektor y> [,[<atrament>][,tryb rysowania>]]

```
10 REM Za pomocą klawiszów kursora rysujemy linie
20 BORDER 0:GRAPHICS PEN 1
30 MODE 1:PLOT 320,200
40 IF INKEY (0)=0 THEN PLOTR 0,1
50 IF INKEY (1)=0 THEN PLOTR 1,0
60 IF INKEY (2)=0 THEN PLOTR 0,-1
70 IF INKEY (8)=0 THEN PLOTR -1,0
80 IF INKEY (9)=0 THEN 30:REM copy = zetrzyj
90 GOTO 40
run
```

ROZKAZ: Rysuje punkt na ekranie graficznym na pozycji określonej przez podanie wektora przesunięcia w stosunku do bieżącej pozycji kursora graficznego. W rozkazie tym może być podany <atrament>, którym punkt ma być rysowany (w zakresie 0 do 15).

Opcjonalny parametr <tryb rysowania> określa rodzaj interakcji atramentu z punktami już narysowanymi na ekranie graficznym. Są 4 tryby rysowania :

0: Normalny

1 XOR (eXclusive OR)

2: AND (i)

3: OR (lub)

Słowa kluczowe pokrewne: GRAPHICS PEN, PLOT

POKE

POKE <wyrażenie określające adr[illegible] <wyrażenie określające liczbe całkowitą>

```
10 FOR m=49152 TO 65535
20 POKE m,100
30 NEXT
run
```

ROZKAZ: Wpisuje wartość wyrażenia określającego liczbę całkowitą (w zakresie 0 do 255) bezpośrednio do pamięci RAM do komórki o wskazanym adresie.

Rozkaz ten nie powinien być stosowany przez niedoświadczonego programistę.

Słowa kluczowe pokrewne: PEEK

POS

POS (#<wyrażenie określające strumień>)

```
 10 MODE 1:BORDER 0:LOCATE 8,2
 20 PRINT "naciskaj klawisze kursora lewo/prawo"
 30 WINDOW 1,40,12,12:CURSOR 1,1
 40 FOR n=1 TO 19:PRINT CHR$ (9);:NEXT
 50 IN INKEY (1) <> -1 THEN PRINT CHR$ (9);
 60 IF INKEY (8) <> -1 THEN PRINT CHR$ (8);
 70 LOCATE # 1,2,24
 80 PRINT # 1,"pozycja horyzontalna";
 90 PRINT # 1,"kursora tekstu =";
100 PRINT #1, POS ( #0): GOTO 50
run
```

FUNKCJA: Podaje aktualną, horyzontalną pozycję (POSition) kursora tekstu w stosunku do lewego krańca okna tekstowego.

Wyrażenie określające strumień MUSI być umieszczone w rozkazie.

POS (# 8) podaje aktualną horyzontalną pozycję karetki drukarki (1 jest pozycją skrajnie lewą).

POS (# 9) podaje pozycję logiczną w strumieniu zbioru dyskowego, to jest liczbą znaków wysłanych licząc od ostatniego znaku CR (&0D).

Słowa kluczowe pokrewne: VPOS, WINDOW

PRINT

PRINT [#<wyrażenie określające strumień>,][<lista: pozycji do druku>]

```
10 a$ = "mały"
20 b$ = "ten ciąg jest nieco większy"
30 PRINT a$; a$
40 PRINT a$, a$
50 PRINT
60 PRINT b$; b$
70 PRINT b$, b$
run
```

ROZKAZ: Drukuje (PRINT) listy: pozycji do druku w danym strumieniu (#0 w braku określenia).

Uwaga. W przypadku użycia średnika (;), który powoduje, że następna <pozycja do druku> jest drukowana obok poprzedniej, BASIC sprawdza najpierw, czy następna pozycja do druku mieści się w tej samej linii. Jeżeli nie, drukowana jest ona w nowej linii, pomimo średnika.

Uwaga. W przypadku użycia przecinka (,), który powoduje, że następna pozycja do druku jest drukowana w następnej strefie tabulacji, BASIC sprawdza najpierw, czy poprzednia pozycja nie przekracza bieżącej strefy tabulacji. Jeżeli tak, następna <pozycja do druku> jest drukowana w dalszej strefie.

PRINT SPC

PRINT TAB

PRINT [#<wyrażenie określające strumień> ,][< lista: < pozycji do druku>] [;] [SPC (<wyrażenie określające liczbę całkowitą>)][<lista: <pozycji do druku>]

PRINT [#<wyrażenie określające strumień> ,][< lista: < pozycji do druku>] [;] [TAB (<wyrażenie określające liczbę całkowitą>)] [<lista: < pozycji do druku>]

```
10 PRINT "tak działa funkcja SPC"
20 FOR x=6 TO 15
30 PRINT SPC (5) "a"; SPC (x) "b"
40 NEXT
50 PRINT "tak działa funkcja TAB"
60 FOR n=6 TO 15
70 PRINT TAB (5) "a"; TAB (x) "b"
80 NEXT
run
```

SPC wpisuje tyle spacji (SPaCe), ile wynosi <wyrażenie określające liczbę całkowitą>. Następna pozycja z listy drukowana jest bezpośrednio za spacjami (zakładając, że następna pozycja mieści się w linii). Stąd kończenie SPC średnikiem nie jest potrzebne.

TAB (TABulacja) wpisuje spacje odliczając od lewego brzegu okna tekstowego tyle znaków, ile wynosi <wyrażenie określające liczbę całkowitą>. Następna pozycja z listy drukowana jest bezpośrednio za spacjami (zakładając, że mieści się w linii). Stąd kończenie TAB średnikiem nie jest potrzebne. Jeżeli bieżąca pozycja drukowanego znaku jest większa od wartości<wyrażenia określającego liczbę całkowitą> następuje przejście do nowej linii i przesunięcie do pozycji wyspecyfikowanej.

PRINT USING

PRINT [# <wyrażenie określające strumień>][< lista:< pozycji do druku>] [;] [USING <szablon formatu>][< separator wyrażenia>]

```
10 FOR x=1 TO 10
20 n=100000 *(RND 5)
30 PRINT "materiały"; USING '########,.##';n
40 NEXT
run
```

PRINT USING (drukuj stosując) pozwala wyspecyfikować format wydruku wyrażenia, kóóre ma być drukowane. Specyfikacja polega na podaniu <szablonu formatu> oddzielonego od wyrażenia <separatorem> w postaci średnika lub przecinka. <Szablon formatu> jest ciągiem tekstowym tworzonym z następujących "specyfikatorów pola formatu":

Formaty liczbowe

Dotyczące samych liczb:

Każdy znak # określa jedną cyfrę

. Przykładowy szablon: #### ##

. Określa miejsce kropki dziesiętnej.
Przykładowy szablon: ######.##

, (Określa jedną cyfrę). Może występować tylko bezpośrednio PRZED kropką dziesiętną. Oznacza, że cyfry przed kropką dziesiętną mają być podzielone na grupy po trzy cyfry oddzielone przecinkami.
Przykładowy szablon: ######,.##

Dotyczące sąsiedztwa liczb:

££ (Określa dwie cyfry.) Oznacza, że bezpośrednio przed pierwszą cyfrą lub kropką dziesiętną (po ewentualnym znaku + lub -), ma być drukowany znak £. Uwaga, £ zajmuje jedno miejsce cyfrowe.
Przykładowy szablon: ££######,.##

** (Określa dwie cyfry.) Oznacza, że ewentualne spacje z przodu liczby mają być zastąpione gwiazdkami
Przykładowy szablon: **######,.##

**£ (Określa trzy cyfry.) Kombinacja ** i ££, to jest: gwiazdki z przodu i znak £
Przykładowy szablon: **£#####,.##

$$ (Określa dwie cyfry.) Oznacza, że bezpośrednio przed pierwszą cyfrą lub kropką dziesiętną (po ewentualnym znaku + lub -) ma być drukowany znak $. Uwaga, $ zajmuje jedno miejsce cyfrowe.
Przykładowy szablon: $$#####,.##

**$ (Określa trzy cyfry.) Kombinacja ** i $$, to jest: gwiazdki z przodu i znak $

\+ Oznacza, że ma być drukowany znak liczby (+ dla dodatnich, - dla ujemnych). Znak - dla liczb ujemnych jest, w braku określenia, drukowany z przodu liczby. Jeżeli + jest umieszczony z przodu szablonu, znak (+ lub -) jest drukowany bezpośrednio przed liczbą (i ewentualnym oznaczeniem waluty - £ lub $). Jeżeli + jest umieszczony na końcu szablonu, znak drukowany jest za liczbą (i ewentualnym wykładnikiem)
Przykładowy szablon: **$#####,.##+

— Może być umieszczony wyłącznie na końcu szablonu. Oznacza, że znak - ma być drukowany za liczbą ujemną (i wykładnikiem). Jeżeli liczba jest dodatnia, drukowana jest spacja. Znak - dla liczb ujemnych jest, w braku określenia, drukowany przed liczbą, chyba, że szablon formatu określa inaczej.

Przykładowy szablon: ####.#### -

↑↑↑↑ Oznacza, że liczba ma być drukowana w postaci wykładniczej. ↑↑↑↑ w szablonie powinno zajmować miejsce PO znakach określających cyfry, ale PRZED ewentualnym znakiem + lub - na końcu szablonu.

Przykładowy szablon: #.#### ↑↑↑↑+

<Szablon formatu> dla liczb nie może przekraczać 20 znaków. Liczby są zaokrąglane do liczby miejsc określonych przez szablon.

Jeżeli szablon formatu jest za mały dla wyrażenia wejściowego, na przykład:

```
PRINT USING "####"; 12345678
```

... wydruk NIF jest skracany do długości szablonu, lecz jest poprzedzony znakiem % oznaczającym uchybienie formatu.

Formaty tekstowe

```
10 CLS:a$="abcdefghijklmnopqrst"
20 PRINT "wyrażenie wejściowe=";a$
30 PRINT:PRINT "specyfikator != ";
40 PRINT USING "!";a$
50 PRINT:PRINT "specyfikator \spacje\ =";
60 PRINT USING "\   \"; a$
70 PRINT:PRINT "specyfikator &= ";
80 PRINT USING "&";a$
90 GOTO 90
run
```

! Oznacza, że drukowany ma być tylko pierwszy znak tekstu

Przykładowy szablon: !

\<spacje>\

Oznacza, że drukowane ma być x pierwszych znaków tekstu, gdzie x jest równe długości szablonu (wliczając kreski).
Przykładowy szablon \   \

& Oznacza, że ma być drukowany cały tekst.
Przykładowy szablon: &

<Szablon formatu> dla tekstu nie może być dłuższy niż 255 znaków. <Szablonyformatu> zarówno dla liczb, jak i dla tekstów mogą występować w postaci zmiennych tekstowych, na przykład:

```
10 a$ ='££######,.##'
20 b$="!"
30 PRINT USING a$; 12345.6789;
40 PRINT USING b$; "groszy"
run
```

Dalsze informacje dotyczące formatowania wydruków można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili..."
Słowa kluczowe pokrewne: SPC, TAB, USING, ZONE

RAD

RAD

```
RAD
```

ROZKAZ: Ustala tryb liczenia kątów w RADianach. BASIC jest ustawiony na radiany po włączeniu do sieci, po wyzerowaniu komputera oraz po rozkazach NEW, CLEAR, LOAD, RUN itp.
Słowa kluczowe pokrewne: ATN, COS, DEG, SIN, TAN

RANDOMIZE

RANDOMIZE [<wyrażenie liczbowe>]

```
RANDOMIZE 123.456
PRINT RND
0.258852139
```

ROZKAZ: Ustanawia wartość początkową funkcji RND określoną wyrażeniem liczbowym . Generator liczb losowych BASIC'a daje pseudolosową sekwencję, w której każda liczba zależy od poprzedniej liczby, zaczynając od danej liczby początkowej. Sekwencja jest zawsze taka sama. RANDOMIZE ustala nową wartość początkową

dla generatora liczb losowych, równą wartości <wyrażenia liczbowego>, lub, jeżeli <wyrażenie liczbowe> jest pominięte, równą wartości wprowadzonej przez użytkownika.
RANDOMIZE TIME daje sekwencję trudną do powtórzenia.
Słowa kluczowe pokrewne: RND

READ

READ <lista :< zmiennych>

```
10 FOR n=1 TO 8
20 READ a$,c
30 PRINT a$;" ";:SOUND 1,c:NEXT
40 DATA to,478,jest,426,8,379,dźwięków
50 DATA 358,muzycznej,319,skali,284,C,253,Dur,239
run
```

ROZKAZ: Czyta (READ) dane ze zbiorów DATA i podstawie je do zmiennych, automatycznie przesuwając "wskaźnik" w zbiorze DATA na następną pozycję. Rozkaz RESTORE powoduje powrót "wskaźnika" na początek zbioru DATA.

Dalsze informacje dotyczące danych można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: DATA, RESTORE

RELEASE

RELEASE <kanały dźwiękowe>

```
10 SOUND 65,1000,100
20 PRINT "Naciśnij R, żeby zakończyć dźwięk"
30 IF INKEY (50) =-1 THEN 30
40 RELEASE 1
run
```

ROZKAZ: Zwalnia (RELEASE) kanały dźwiękowe, które są w stanie podtrzymania po rozkazie SOUND

Parametr <kanały dźwiękowe> musi mieć wartość całkowitą w zakresie 1 do 7. Wartości te powodują:
1: Zwalnia kanał A
2: Zwalnia kanał B
3: Zwalnia kanał A i B

4: Zwalnia kanał C
5: Zwalnia kanał A i C
6: Zwalnia kanał B i C
7: Zwalnia kanał A, B i C
Dalsze informacje dotyczące dźwięku można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: SOUND

REM

REM <reszta linii>

```
10 REM Mega-monstra, najeźdźcy z intergalaktycznej
   hiperprzestrzeni. Gra opracowana przez MASOFT
20 REM Copyright AMSOFT 1985
```

ROZKAZ: Wprowadza komentarze (REMark) do programu. <Reszta linii> jest ignorowana przez BASIC i może zawierać dowolne znaki, również dwukropki :, które normalnie odzielają rozkazy.

Zamiast REM można używać znaku pojedynczego cudzysłowu we wszystkich zastosowaniach z wyjątkiem instrukcji DATA gdzie jest traktowane jako część ciągu tekstowego.
Słowa kluczowe pokrewne: nie ma

REMAIN

REMAIN (<numer licznika>)

```
10 AFTER 500,1 GOSUB 40
20 AFTER 100,2 GOSUB 50
30 PRINT "program jest wykonywany":GOTO 30
40 REM ten podprogram nie będzie wywołany, ponieważ
   jest zakazany w linii 80
50 PRINT:PRINT "Licznik 1 będzie teraz ";
60 PRINT "zatrzymany przez REMAIN."
70 PRINT "Bozostało";
80 PRINT REMAIN (1);
90 PRINT "jednostek czasu."
run
```

FUNKCJA: Podaje pozostałą (REMAINing) zawartość licznika o danym numerze (w zakresie 0 do 3) i zatrzymuje licznik.

Dalsze informacje dotyczące przerwań można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: AFTER, DI, EI, EVERY

RENUM

RENUM [<nowy numer linii>][,[<stary numer linii>] [,<inkrement>]]

```
10 CLS
20 REM to będzie linia 123
30 REM to będzie linia 124
40 REM to będzie linią 125
RENUM 123,20,1
LIST
```

ROZKAZ: Powoduje przenumerowanie (RENUMbering) linii programu. Parametr <stary numer linii> określa aktualnie istniejącą linię, od której ma się rozpocząć przenumerowanie. Jeżeli <stary numer linii> jest pominięty, przenumerowane zostaną linie od początku programu

Parametr <nowy numer linii> określa nowy numer linii, od której rozpoczyna się przenumerowanie. Jeżeli parametr ten jest pominięty, przenumerowany program zaczyna się od linii 10.

<Inkrement> określa krok pomiędzy przenumerowanymi liniami. Jeżeli parametr ten jest pominięty, przenumerowane linie różnią się o 10.

RENUM uwzględnia GOSUB, GOTO i inne odwołania do numerów linii. Nie zmieniane są natomiast odniesienia do numerów linii wewnątrz wyrażeń tekstowych (na przykład w rozkazach KEY), ani odniesienia w komentarzach REM, ani < wyrażenia określające numer linii > w rozkazach CHAIN lub CHAIN MERGE.
Numery linii mogą mieć wartości od 1 do 65535.
Słowa kluczowe pokrewne: DELETE, LIST

RESTORE

RESTORE <numer linii>

```
10 READ a$:PRINT a$;" ";
20 RESTORE 50
30 FOR t=1 TO 500:NEXT:GOTO 10
40 DATA odtworzone dane mogą być czytane
50 DATA od nowa
run
```

ROZKAZ: Odtwarza (RESTORE) początkową pozycję "wskaźnika" linii DATA o numerze podanym opcjonalnie. Jeżeli numer linii nie jest podany, rozkaz przesuwa "wskaźnik" na początek pierwszej linii danych.

Dalsze informacje dotyczące danych można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: DATA, READ

RESUME

RESUME [< numer linii>]

```
10 ON ERROR GOTO 60
20 FOR x=10 TO 0 STEP-1:PRINT 1/x:NEXT
30 END
40 PRINT "chodź tutaj po stwierdzeniu błędu"
50 END
60 PRINT "błąd nr."; ERR; "w linii"; ERL
70 RESUME 40
run
```

ROZKAZ: Przywraca (RESUME) normalne wykonywanie programu po stwierdzeniu i opracowaniu przez rozkaz ON ERROR GOTO błędu. Jeżeli <numer linii> nie jest podany, wykonywanie programu rozpoczyna się na nowo od linii, w której błąd został wykryty. Spróbuj usunąć < numer linii> :

```
70 RESUME
run
```

Słowa kluczowe pokrewne: ERR, ERL, ERROR, ON ERROR GOTO, RESUME NEXT

RESUME NEXT

RESUME NEXT

```
10 ON ERROR GOTO 90
20 PRINT "za każdym razem naciśnij RETURN
30 INPUT "1";a
40 INPUT "2";a
50 inpot "3"; a:REM błąd składni!
60 INPUT "4";a
```

```
 70 INPUT "5";a
 80 END
 90 PRINT "błąd nr."; ERR; "w linii"; ERL
100 RESUME NEXT
run
```

ROZKAZ: Przywraca (RESUME) normalne wykonywanie programu po stwierdzeniu i opracowaniu przez rozkaz ON ERROR GOTO błędu

RESUME NEXT powoduje powtórne rozpoczęcie wykonywania programu od linii następnej (NEXT) za linią, w której błąd został wykryty.

Słowa kluczowe pokrewne: DERR, ERL, ERROR, ON ERROR GOTO, RESUME

RETURN

RETURN

```
10 GOSUB 50:PRINT "już po podprogramie":END
50 FOR n=1 TO 20
60 PRINT "podprogram"
70 NEXT:PRINT
80 RETURN
run
```

ROZKAZ: Zaznacza koniec podprogramu BASIC powraca (RETURN) z podprogramu do instrukcji bezpośrednio za rozkazem GOSUB, który wywołał podprogram.

Słowa kluczowe pokrewne: GOSUB

RIGHT$

RIGHT$ (<wyrażenie tekstowe> , <żądana długość>)

```
10 MODE 1:a$="CPC6128 computer"
20 FOR n=1 TO 16:LOCATE 41-n,n
30 PRINT RIGHT$ (a$,n)
40 NEXT
run
```

FUNKCJA: Podaje ciąg znaków tekstowych o długości (w zakresie 0 do 255) określonej przez parametr <żądana długość> , który jest prawą (RIGHT) częścią <wyrażenia tekstowego> . Jeżeli

<wyrażenie tekstowe> jest krótsze niż <żądana długość> , podawane jest całe wyrażenie tekstowe
Słowa kluczowe pokrewne: LEFT$, MID$

RND

RND [(< wyrażenie liczbowe >)]

```
10 RANDOMIZE
20 FOR x=1 TO -1 STEP -1
30 PRINT "parametr rnd=";x
40 FOR n=1 TO 6
50 PRINT RND (x)
60 NEXT n,x
run
```

FUNKCJA: Podaje następną liczbę z sekwencji liczb losowych (RaNDom), jeżeli <wyrażenie liczbowe> jest dodatnie lub jeżeli nie jest specyfikowane.

Jeżeli <wyrażenie liczbowe> ma wartość zero, RND podaje kopię ostatniej wygenerowanej liczby losowej.

Jeżeli <wyrażenie liczbowe> ma wartość ujemną, rozpoczynana jest nowa sekwencja liczb losowych, z której podawana jest pierwsza liczba.

Słowa kluczowe pokrewne: RANDOMIZE

ROUND

ROUND (<wyrażenie liczbowe> [, < miejsca dziesiętne])

```
10 FOR n=4 TO -4 STEP -1
20 PRINT ROUND (1234.5678,n),
30 PRINT "z dokładnością do"; n;"miejsc dziesiętnych"
40 NEXT
run
```

FUNKCJA: Zaokrągla wyrażenie liczbowe do liczby <miejsc dziesiętnych> . Jeżeli parametr <miejsca dziesiętne> jest ujemny, liczba jest zaokrąglana tak, że przed kropką dziesiętną wpisywana jest określona miejscami dziesiętnymi liczba zer.

Słowa kluczowe pokrewne: ABS, CINT, FIX, INT

RUN

RUN <wyrażenie tekstowe>

    RUN "disc"

ROZKAZ: Laduje program typu BASIC lub binarny z dysku i rozpoczyna wykonywanie. Jakikolwiek wcześniej załadowany program jest usuwany z pamięci.

Programy zabezpieczone mogą być uruchamiane w ten sposób.

Słowa kluczowe pokrewne: LOAD

RUN

RUN [<numer linii>]

    RUN 200

ROZKAZ: Rozpoczyna wykonywanie programu, napisanego w języku BASIC i umieszczonego w pamięci, od linii o wskazanym numerze, lub od początku, jeżeli numer linii nie jest podany. Rozkaz RUN powoduje wyzerowanie wszystkich zmiennych.

Programy zabezpieczone nie mogą być uruchamiane w ten sposób (po załadowaniu).

Słowa kluczowe pokrewne: CONT, END, STOP

SAVE

SAVE <nazwa zbioru> [, <typ zbioru>] [, <parametry binarne>]

    SAVE "discfile.xyz"

... zapisuje na dysku aktualny program pamięci jako normalny, niezabezpieczony zbiór typu BASIC.

    SAVE "discfile.xyz",P

... zapisuje na dysku zabezpieczony zbiór typu BASIC

    SAVE "discfile.xyz",A

... zapisuje na dysku zbiór typu ASCII.

    SAVE "discfile.xyz",B,8000,3000,8001

... zapisuje na dysku zbiór binarny. W tym przykładzie zapisuje na dysku zawartość pamięci od adresu 8000, przy czym długość zbioru wynosi 3000 bajtów, 8001 jest opcjonalnym adresem wejściowym, od którego program ma być uruchamiany.

ROZKAZ: Zapisuje program z pamięci na dysk Zbiór Binarny jest częścią pamięci przepisaną na dysk. Parametry Binarne są następujące:

<adres początkowy> , <długość zbioru> , <adres wejściowy>

Pamięć ekranu może być również zapisana na dysku jako zbiór Binarny. Znane jest to jako screen dump i może być wykonane rozkazem:

```
SAVE "ekran", B, &C000, &4000
```

W celu załadowania zbioru z powrotem do pamięci ekranu:

```
LOAD "ekran"
```

Słowa kluczowe pokrewne: CHAIN, CHAIN MERGE, LOAD, MERGE, RUN

SGN

SGN (<wyrażenie liczbowe>)

```
10 FOR n=200 TO -200 STEP -20
20 PRINT "dla wartości";n
30 PRINT "SGN daje wynik:"; SGN (n)
40 NEXT
run
```

FUNKCJA: Podaje znak (SIGN) <wyrażenia liczbowego> . Wynik jest równy -1 dla liczb ujemnych, 0 dla wartości zero i 1 dla liczb dodatnich.

Słowa kluczowe pokrewne: ABS

SIN

SIN (<wyrażenie liczbowe>)

```
10 CLS:DEG:ORIGIN 0,200
20 FOR n=0 TO 720
30 y=SIN (n)
40 PLOT n*640/720, 198*y:NEXT
50 GOTO 50
run
```

FUNKCJA: Oblicza SINus <wyrażenia liczbowego> .

Uwaga, rozkazy DEG i RAD przełączają komputer na liczenie kątów odpowiednio w stopniach i radianach.

Słowa kluczowe pokrewne: ATN, COS, DEG, RAD, TAN

SOUND

SOUND <określenie kanału> , <okres tonu>[, <czas trwania>
[,<głośność> [, <obwiednia głośności> [, <obwiednia
tonu> [, <szum>]]]]]

```
10 FOR z=0 TO 4095
20 SOUND 1,z,1,12
30 NEXT
run
```

ROZKAZ: Programuje dźwięk. Rozkaz ten ma następujące parametry:

Parametr 1: <określenie kanału>

Wartość tego parametru musi być liczbą całkowitą w zakresie 1 do 255. Działanie parametru określone jest przez poszczególne bity parametru w sposób następujący:

Bit 0: (dziesiętnie 1) wyślij dźwięk do kanału A (najmniej znaczący bit)
Bit 1: (dziesiętnie 2) wyślij dźwięk do kanału B
Bit 2: (dziesiętnie 4) wyślij dźwięk do kanału C
Bit 3: (dziesiętnie 8) "rendezvous" z kanałem A
Bit 4: (dziesiętnie 16) "rendezvous" z kanałem B
Bit 5: (dziesiętnie 32) "rendezvous" z kanałem C
Bit 6: (dziesiętnie 64) podtrzymanie kanału dźwięku
Bit 7: (dziesiętnie 128) wyzerowanie kanału dźwięku (najbardziej znaczący bit)

Stąd wartość <określenia kanału> równa 68, na przykład, znaczy: Wyślij dźwięk do kanału C(4) z podtrzymaniem (64)

Parametr 2: <okres tonu>

Parametr ten określa wysokość dźwięku lub innymi słowami "jaka to nuta" (to jest: do,re,mi,fa,so itd.). Każdej nucie przypisana jest liczba, czyli właśnie <okres tonu>. Zobacz w rozdziale zatytułowanym "Nieco użytecznych informacji ..."

Parametr 3: <czas trwania>

Parametr ten określa, jak długo brzmi dźwięk. Wartość parametru określa czas w jednostkach 0,01 sekundowych. Przy pominięciu tego parametru przyjmowana jest wartość 20 odpowiadająca 1/5 sekundy.

Jeżeli <czas trwania> jest równy zero, dźwięk brzmi do końca podanej obwiedni głośności.

Jeżeli <czas trwania> jest ujemny, podana obwiednia głośności ma być powtarzana ABS (<czas trwania>) razy.

Parametr 4: <głośność>

Parametr ten określa początkową głośność dźwięku. Wartość <głośności> powinna być od 0 do 16. Głośność 0 oznacza wyłączenie głosu, 15 - maksimum. W przypadku braku tego parametru przyjmowana jest wartość 12.

Parametr 5: <obwiednia głośności>

Jeżeli głośność ma zmieniać się podczas trwania dźwięku, należy podać numer <obwiedni głośności> utworzonej oddzielnym rozkazem ENV. W praktyce można utworzyć do 15 różnych obwiedni głośności o numerach od 1 do 15. Podanie numeru <obwiedni głośności> w rozkazie SOUND powoduje wywołanie odpowiedniej obwiedni.
Zobacz opis rozkazu ENV.

Parametr 6: <obwiednia tonu>

Jeżeli wysokość tonu ma się zmieniać podczas trwania dźwięku, nalezy podać numer <obwiedni tonu> utworzonej oddzielnym rozkazem ENT. W praktyce można utworzyć do 15 różnych obwiedni tonu o numerach od 1 do 15. Podanie numeru <obwiedni tonu> w rozkazie SOUND powoduje wywołanie odpowiedniej obwiedni. Jeżeli, w rozkazie ENT, numer obwiedni został określony jako ujemny, parametr <obwiednia tonu> w rozkazie SOUND powinien być równy wartości bezwzględnej tego numeru (to jest bez znaku minus).
Zobacz opis rozkazu ENT.

Parametr 7: <szum>

Do dźwięku może być dodany szum biały o zakresie widma zmienianym wartością parametru <szum> . Wartość ta powinna być w zakresie 0 do 31.

Dalsze informacje dotyczące dźwięku można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ..."
Słowa kluczowe pokrewne: ENT, ENV, ON SQ GOSUB, RELEASE, SQ

SPACE$

SPACE$ (<wyrażenie określające liczbę całkowitą>)

```
10 MODE 1
20 PRINT "Daj 9 spacji pomiędzy sobą";
30 PRINT SPACE$ (9);
40 PRINT "i mną!"
run
```

FUNKCJA: Tworzy ciąg spacji o długości (w zakresie 0 do 255) podanej w <wyrażeniu określającym liczbę całkowitą> .
Słowa kluczowe pokrewne: SPC, STRING$, TAB

SPC

(Patrz PRINT SPC)

SPEED INK

SPEED INK <okres 1>, < okres 2>

```
10 BORDER 7,18
20 FOR 1=30 TO 1 STEP -1
30 SPEED INK 1,k
40 FOR t=1 TO 700:NEXT t,1
run
```

ROZKAZ: Ustala jak długo atrament ma dany kolor (dotyczy atramentów o dwóch przełączanych kolorach określonych rozkazem INK lub BORDER). <Okres 1> określa czas (w jednostkach 0,02 sekundowych) dla pierwszego koloru, <okres 2> - dla drugiego koloru.

Należy uważać, przy doborze kolorów przełączanych i ich okresów, ponieważ mogą wystąpić efekty hipnotyczne.
Słowa kluczowe pokrewne: BORDER, INK

SPEED KEY

SPEED KEY <opóźnienie startu> , < okres powtarzania>

```
10 CLS:FOR k=7 TO 1 STEP -2
20 PRINT "wpisz swoje imię, następnie [RETURN]"
30 SPEED KEY k,k
40 LINE INPUT a$:NEXT
50 PRINT "Śmieszne imię!"
run
```

ROZKAZ: Ustala czas automatycznego powtarzania zgłoszenia naciśnięcia klawisza. Parametr <opóźnienie startu> określa czas (w jednostkach 0,02 sekundowych) pomiędzy naciśnięciem klawisza, a wystartowaniem automatycznego powtarzania. Parametr <okres powtarzania> określa czas pomiędzy kolejnymi zgłoszeniami danego klawisza.

SPEED KEY działa tylko na klawisze, które mają automatyczne powtarzanie ustawione domyślnie, lub którym ustawiono automatyczne powtarzanie rozkazem KEY DEF.

W przypadku używania małych wartości <opóźnienia startu>, rozsądnym jest wstępne zaprogramowanie jednego z klawiszów cyfrowych 0÷9 na powrót do "normalnych parametrów" 30,2. Można to osiągnąć rozkazem:

```
KEY 0, "SPEED KEY 30,2"+CHR$(13)
```

... który ustawi parametry początkowe po naciśnięciu 0

Słowa kluczowe pokrewne: KEY DEF

SPEED WRITE

SPEED WRITE < wyrażenie określające liczbę całkowitą >

```
SPEED WRITE 1
```

ROZKAZ: Określa szybkość zapisu na taśmie magnetofonowej (jeżeli dołączony jest magnetofon). Szybkość zapisu na taśmie jest równa 2000 bodów (bitów na sekundę), jeżeli wartość < wyrażenia określającego liczbę całkowitą > wynosi 1, lub 1000 bodów (szybkość ustawiana w braku określenia), jeżeli wartość < wyrażenia > jest równa 0. Przy ładowaniu z taśmy magnetofonowej komputer automatycznie wybiera właściwą szybkość odczytu.

Dla zwiększenia niezawddności zapisu-odczytu danych poleca się używać SPEED WRITE 0 (ustawiane automatycznie w braku określenia).

Rozkaz SPEED WRITE nie ma wpływu na operacje z dyskami.
Słowa kluczowe pokrewne: OPENOUT, SAVE

SQ

SQ (<kanał>)

```
10 SOUND 65,100,100
20 PRINT SQ (1)
run
67
```

FUNKCJA: Podaje stan kolejki dźwiękowej (Sound Queue) dla danego <kanału> . Kanał musi być wyrażeniem dającym jedną z wartości:

1: dla kanału A
2: dla kanału B
4: dla kanału C

Wynikiem funkcji SQ jest liczba całkowita, której bity postaci binarnej oznaczają:

Bity 0, 1 i 2: liczba wolnych miejsc w kolejce
Bity 3, 4 i 5: stan "rendezvous" pierwszego dźwięku w kolejce
Bit 6: stan podtrzymania pierwszego dźwięku w kolejce
Bit 7: stan aktywny kanału

... gdzie Bit 0 jest najmniej znaczący, Bit 7 najbardziej znaczący.

Z powyższego określenia wynika, że, jeżeli Bit 6 jest 1, Bit 7 nie może być 1 i odwrotnie. Podobnie, jeżeli Bit 3, 4 lub 5 jest 1, Bity 6 i 7 nie mogą być 1.

Dalsze informacje dotyczące dźwięku można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ...".

Słowa kluczowe pokrewne: ON SQ GOSUB, SOUND

SQR

SQR (<wyrażenie liczbowe>)

```
PRINT SQR (9)
3
```

FUNKCJA: Podaje wartość pierwiastka kwadratowego (SQuare Root) <wyrażenia liczbowego>
Słowa kluczowe pokrewne: nie ma

STEP

(Patrz FOR)

STOP

STOP

```
10 FOR n=1 TO 30: PRINT n:NEXT
20 STOP
30 FOR n=31 TO 60:PRINT n:NEXT
run
cont
```

ROZKAZ: Zatrzymuje wykonywanie programu, ale pozostawia BASIC w stanie, który umożliwia jego kontynuację rozkazem CONT. STOP może być używany do zatrzymywania programu w określonym miejscu przy jego uruchamianiu.
Słowa kluczowe pokrewne: CONT, END

STR$

STR$ (<wyrażenie liczbowe>)

```
10 a=&FF:REM 255 hex
20 b=&X1111:REM 15 binarnie
30 c$="***"
40 PRINT c$+STR$(a+b)+c$
run
***270***
```

FUNKCJA: Przetwarza wyrażenie liczbowe na ciąg cyfr dziesiętnych.
Słowa kluczowe pokrewne: BIN$, DEC$, HEX$, VAL

STRING$

STRING$ (<długość> , <specyfikator znaku>)

```
PRINT STRING$ (40,"*")
****************************************
```

FUNKCJA: Daje wyrażenie tekstowe składające się z wyspecyfikowanego znaku powtórzonego <długość> -krotnie (od 0 do 255). Uwaga, powyższy przykład może być napisany również:

```
PRINT STRING$ (40,42)
```

... gdzie <specyfikator znaku> 42 odpowiada wartości ASCII znaku *. Jest to równoważne zapisowi PRINT STRING$ (40,CHR$(42))
Słowa kluczowe pokrewne: SPACE$

SWAP

(Patrz WINDOW SWAP)

SYMBOL

SYMBOL <numer znaku>, <lista: <rzędów>

```
10 MODE 1:SYMBOL AFTER 105
20 rząd 1=255:REM binarnie 1 1 1 1 1 1 1 1
30 rząd 2=129:REM binarnie 1 0 0 0 0 0 0 1
40 rząd 3=189:REM binarnie 1 0 1 1 1 1 0 1
50 rząd 4=153:REM binarnie 1 0 0 1 1 0 0 1
60 rząd 5=153:REM binarnie 1 0 0 1 1 0 0 1
70 rząd 6=189:REM binarnie 1 0 1 1 1 1 0 1
80 rząd 7=129:REM binarnie 1 0 0 0 0 0 0 1
90 rząd 8=255:REM binarnie 1 1 1 1 1 1 1 1
100 PRINT "Linia 110 redefiniuje literę "i" (105)
    Wpisz kilka liter "i" i popatrz! Potem wylistuj
    program"
110 SYMBOL 105, rząd 1, rząd 2, rząd 3, rząd 4, rząd 5,
    rząd 6, rząd 7, rząd 8
run
```

ROZKAZ: Redefiniuje kształt znaku na ekranie. Każdy z parametrów musi dawać liczbę całkowitą w zakresie 0 do 255.

Dla nowo definiowanego znaku komputer musi przeznaczyć wpierw miejsce w pamięci. Czyni się to rozkazem

```
SYMBOL AFTER x
```

... gdzie x jest równe lub mniejsze od numeru znaku, który ma być redefiniowany.

Następnie wydaje się rozkaz SYMBOL, którego pierwszym parametrem jest numer znaku x.

Bez względu na to, czy wartość x określa znak wpisywany bezpośrednio z klawiatury, czy nie, nowo zdefiniowany znak może być wydrukowany na ekranie rozkazem:

PRINT CHR$(x)

Po SYMBOL x, występuje do 8 parametrów, które określają indywidualnie 8 kolejnych, licząc od góry, poziomych rzędów znaku. Każdy z parametrów może być w zakresie 0 do 255. Postać binarna każdego parametru określa wzór danego rzędu znaku.

Na przykład, jeżeli pierwszy z 8 parametrów jest równy 1, to wzór najwyższego rzędu jest określony liczbą binarną 00000001. Rząd podzielony jest na 8 sekcji odpowiadających ośmiu bitom liczby binarnej. Jeżeli na pozycji danej sekcji jest 1, sekcja ta jest drukowana w kolorze pióra. Jeżeli na pozycji danej sekcji jest 0, sekcja ta jest drukowana w kolorze papieru (nie jest widzialna). W przykładzie najwyższy rząd ma kropkę w prawej skrajnej sekcji. Kontynuując przykład określamy następne 7 parametrów jako 3, 7, 15, 31, 63, 0, 0. Postaci binarne tych 8 parametrów są następujące:

parametr (rząd) 1:00000001 binarnie: (1 dziesiętnie)
parametr (rząd) 2:00000011 binarnie: (3 dziesiętnie)
parametr (rząd) 3:00000111 binarnie: (7 dziesiętnie)
parametr (rząd) 4:00001111 binarnie: (15 dziesiętnie)
parametr (rząd) 5:00011111 binarnie: (31 dziesiętnie)
parametr (rząd) 6:00111111 binarnie: (63 dziesiętnie)
parametr (rząd) 7:00000000 binarnie: (0 dziesiętnie)
parametr (rząd) 8:00000000 binarnie: (0 dziesiętnie)

Patrząc na rozkład zer i jedynek w powyższym przykładzie można określić kształt nowego znaku. Przypiszmy te parametry znakowi o numerze 255 rozkazem:

SYMBOL 255,1,3,7,15,31,63,0,0

Uwaga, wartość 0 ostatnich dwóch parametrów pozwala na skrócenie zapisu:

SYMBOL 225,1,3,7,15,31,63

Uwaga, parametry mogą być zapisane w postaci binarnej, co eliminuje konieczność konwersji zaprojektowanego wzoru znaku na liczby dziesiętne (należy pamiętać o przedrostku X oznaczahącym liczby binarne). Na przykład:

```
SYMBOL 255, &X00000001, &X00000011, &X00000111,
&X0000111, & X00011111, &X00111111
```

... Teraz, żeby zobaczyć znak:

```
PRINT CHR$ (255)
```

Przypisanie powyższych parametrów znakowi wpisywanemu z klawiatury powoduje, że nowy znak pojawia się przy każdym naciśnięciu na właściwy klawisz oraz przy każdej próbie wydrukowania starego znaku.

Dalsze informacje dotyczące znaków definiowanych przez użytkownika można znaleźć w 2 części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: SYMBOL AFTER

SYMBOL AFTER

SYMBOL AFTER <wyrażenie określające liczbę całkowitą>

```
10 CLS
20 SYMBOL AFTER 115
30 PRINT "Linia 40 redefiniuje s";
40 SYMBOL 115,0,56,64,48,8,8,112
50 PRINT "na s"
60 PRINT "Usuń tę definicję s, pisząc: SYMBOL AFTER 240"
run
```

ROZKAZ: Określa liczbę znaków, które mogą być definiowane przez użytkownika (w zakresie 0 do 256). Wstępnie wartość ta wynosi 240, co daje 16 znaków, które mogą być definiowane przez użytkownika (od 240 do 255). Jeżeli <wyrażenie> ma wartość 32, wszystkie znaki od 32 do 255 mogą być redefiniowane. SYMBOL AFTER 256 blokuje możliwość zmiany wszystkich znaków.

Przy każdym wykonaniu rozkazu SYMBOL AFTER, zdefiniowane przez użytkownika znaki są kasowane do postaci początkowej.

SYMBOL AFTER NIE funkcjonuje PO zmianie wartości HIMEM rozkazem MEMORY oraz PO otwarciu bufora zbiorów rozkazami OPENIN lub OPENOUT. W takich okolicznościach sygnalizowany jest błąd (5),

"Improper argument" (chyba, że poprzedni stan był SYMBOL AFTER 256).

Dalsze informacje dotyczące znaków definiowanych przez użytkownika można znaleźć w rozdziale zatytułowanym "W wolnej chwili ..."

Słowa kluczowe pokrewne: HIMEM, MEMORY, SYMBOL

TAB

(Patrz PRINT TAB)

TAG

TAG [#<wyrażenie określające strumień>]

```
10 INPUT "napisz swoje imię"; a$:CLS
20 PRINT "Trzymaj się"; a$
30 TAG
40 x=LEN (a$)  17:y=50+RND *300:MOVE -x,y
50 FOR f=-x TO 640 STEP RND  7+3
60 MOVE f,y:PRINT " ";a$;:FRAME:NEXT
70 FOR b=640 TO -x STEP-RND  7+3
80 MOVE b,y:PRINT a$;" ";:FRAME:NEXT
90 GOTO 40
run
```

ROZKAZ: Powoduje, że tekst wysłany do danego strumienia jest drukowany od pozycji kursora graficznego (Text At Graphics). Pozwala to na mieszanie tekstu i symboli znakowych z elementami graficznymi oraz na przemieszczanie znaków "punkt po punkcie" w przeciwieństwie do "znak po znaku". W braku określenia przyjmowany jest strumień #0.

Z pozycją kursora graficznego związany jest lewy, górny róg znaku. Jeżeli instrukcja PRINT nie jest zakończona średnikiem, wyświetlane są również znaki kontrolne (przesunięcie o linię i powrót do początku linii).

Jeżeli rozkau TAG dotyczy strumienia #0, BASIC wyłącza TAG po przejściu do trybu bezpośredniego.

Słowa kluczowe pokrewne: TAGOFF

TAGOFF

TAGOFF [#<wyrażenie określające strumień>]

```
10 MODE 2:TAG:REM Włączono "tekst na ekranie graficznym"
20 rok=1984:FOR x=1 TO 640 STEP 70
30 MOVE x,400:DRAWR 0,-350
40 rok=rok+1:PRINT rok;:NEXT
50 TAGOFF:REM Wyłączono "tekst na ekranie graficznym"
60 LOCATE 34,25:PRINT "Wykresy roczne"
70 GOTO 70
run
```

ROZKAZ: Kasuje TAG dla danego strumienia (strumień #0 w braku określenia), przywraca powiązanie tekstu z kursorem tekstu (pozycja jak przed włączeniem TAG).
Słowa kluczowe pokrewne: TAG

TAN

TAN (<wyrażenie liczbowe>)

```
PRINT TAN (45)
1.61977519
```

FUNKCJA: Oblicza TANgens <wyrażenia liczbowego> , które musi być w zakresie -200000 do +200000
Uwaga, rozkazy DEG i RAD przełączają komputer na liczenie kątów odpowiednio w stopniach i radianach.
Słowa kluczowe pokrewne: ATN, COS, DEG, RAD, SIN

TEST

TEST (<współrzędna x> , < współrzędna y>)

```
10 CLS
20 PRINT "Pióro jest napełnione atramentem numer";
30 PRINT TEST (10,386)
40 PRINT "Zmień atrament pióra i MODE";
50 PRINT "... potem uruchom program jeszcze raz."
run
```

FUNKCJA: Przesuwa kursor graficzny na pozycję określoną współrzędnymi (x,y) i podaje numer atramentu na tej pozycji.
Słowa kluczowe pokrewne: MOVE, MOVER, TESTR, XPOS, YPOS

TESTR

TESTR (<wektor x >, <wektor y>)

```
10 MODE 0:FOR x=1 TO 15:LOCATE 1,x
20 PEN x:PRINT STRING$ (10,143); NEXT
30 MOVE 200, 400:PEN 1
40 FOR N=1 TO 23:LOCATE 12,n
50 PRINT "pióro"; TESTR (0,-16):NEXT
run
```

FUNKCJA: Przesuwa kursor graficzny od pozycji bieżącej o wektor o współrzędnych (x,y) i podaje numer atramentu na tej pozycji.
Słowa kluczowe pokrewne: MOVE, MOVER, TEST, XPOS, YPOS

THEN

(Patrz IF)

TIME

TIME

```
 10 CLS:REM zegar
 20 INPUT "godzina"; godzina
 30 INPUT "minuta"; minuta
 40 INPUT "sekunda"; sekunda
 50 CLS:data=INT (TIME/300)
 60 WHILE godzina<13
 70 WHILE minuta<60
 80 WHILE tik<60
 90 tik=(INT(TIME/300)-data)+ sekunda
100 LOCATE 1,1
110 PRINT USING "##"; godzina, minuta, tik
120 WEND
130 tik=0:sekunda=0:minuta=minuta+1
140 GOTO 50
150 WEND
160 minuta:0:godzina=godzina+1
170 WEND
180 godzina=1
190 GOTO 60
run
```

FUNKCJA: Podaje czas od ostatniego włączenia komputera lub od ostatniego wyzerowania (z wyłączeniem czasu przeznaczonego na obsługę dysków.

Każda sekunda czasu rzeczywistego jest równa wartości: TIME/300

Słowa kluczowe pokrewne: AFTER, EVERY, WEND, WHILE

TO

(Patrz FOR)

TROFF

TRON

TROFF

TRON

```
10 TROFF:PRINT:PRINT "Ślad wyłączony"
20 FOR n=1 TO 8
30 PRINT "program jest wykonywany";NEXT
40 IF f=1 THEN END
50 TRON:PRINT:PRINT "Ślad włączony"
60 f=1:GOTO 20
run
```

ROZKAZ: Włącza podgląd numerów linii w trakcie wykonywania programu. Numer każdej linii drukowany jest, w nawiasach kwadratowych [] , przed jej wykonaniem.

TRON (TRace ON) włącza podgląd, TROFF wyłącza. Udogodnienie to jest szczególnie użyteczne przy śledzeniu przyczyn wystąpienia błędu.

Słowa kluczowe pokrewne: nie ma

UNT

UNT (<wyrażenie określające adres>)

```
PRINT UNT (&FF66)
-154
```

FUNKCJA: Podaje liczbę całkowitą z zakresu -32768 do 32767, która jest odpowiednikiem, w zapisie uzupełnień do dwóch wartości <wyrażenia określającego adres>

Słowa kluczowe pokrewne: CINT, FIX, INT, ROUND

UPPER$

UPPER$ (<wyrażenie tekstowe>)

```
10 CLS:a$="ojej, jak urosłeś!"
20 PRINT UPPER$ (a$)
run
```

FUNKCJA: Podaje nowy tekst, który jest kopią tekstu pierwotnego ale w którym wszystkie małe litery od A do Z są zamienione na duże litery. Funkcja ta jest użyteczna przy przetwarzaniu tekstów wejściowych, które mogą mieć pomieszane litery małe i duże.

Słowa kluczowe pokrewne: LOWER$

USING

(Patrz PRINT USING)

VAL

VAL (<wyrażenie tekstowe>)

```
10 CLS:PRINT "Znam tabliczkę mnożenia!"
20 PRINT:PRINT "Naciśnij klawisz (1 do 9)"
30 a$=INKEY$:IF a$="" THEN 30
40 n=VAL (a$):IF n<1 OR n>9 THEN 30
50 FOR x=1 TO 12
60 PRINT n;"X";x;"=";n*x
70 NEXT:GOTO 20
run
```

FUNKCJA: Podaje wartość (VALue) liczbową (z ewentualnym znakiem i kropką dziesiętną) pierwszego znaku (pierwszych znaków) danego <wyrażenia tekstowego>.

Jeżeli pierwszy znak nie jest cyfrą, wynik jest równy 0. Jeżeli pierwszy znak jest minusem lub kropką dziesiętną, a następny nie jest cyfrą, sygnalizowany jest błąd (13): Type mismatch .

Słowa kluczowe pokrewne: STR$

VPOS

VPOS (#<wyrażenie określające strumień>)

```
10 MODE 1:BORDER 0:LOCATE 8,2
20 PRINT "naciskaj klawisze kursora  góra/dół "
30 WINDOW 39,39,1,25:CURSOR 1,1
40 LOCATE 1,13
50 IF INKEY (0) <> -1 THEN PRINT CHR$ (11);
60 IF INKEY (2) <> -1 THEN PRINT CHR$ (10);
70 LOCATE #1,3,24
80 PRINT # 1,"pozycja pionowa";
90 PRINT # 1,"kursora tekstowego=";
100 PRINT #1,VPOS (#0):GOTO 50
```

FUNKCJA: Podaje bieżącą pozycję pionową (Vertical POSition) kursora tekstowego w odniesieniu do górnej krawędzi okna tekstowego. Wyrażenie określające strumień MUSI być podane, NIE jest przyjmowane domyślnie # 0.
Słowa kluczowe pokrewne: POS, WINDOW

WAIT

WAIT <numer portu>, <maska> [, <inwersja>]

```
WAIT &FF34,20,25
```

ROZKAZ: Zatrzymuje program do czasu odczytania z danego portu określonej wartości z zakresu 0 do 255. Wartość odczytana z portu jest poddawana operacji wyłącznie - lub (exclusive or) z wartością <inwersją>, a następnie operacji i (and) z wartością <maska> . Jeżeli tak otrzymana liczba jest różna od zera, program jest kontynuowany.

Czas czekania (WAIT) BASIC'a jest nieokreślony, aż do zaistnienia żądanych warunków.

Rozkaz ten nie powinien być stosowany przez niedoświadczonego programistę.
Słowa kluczowe pokrewne: INP, OUT

WEND

WEND

```
WEND
```

ROZKAZ: Znaczy koniec fragmentu programu, który ma być wykonywany w pętli WHILE. WEND wybiera automatycznie rozkaz WHILE, z którym ma być skojarzony.
Słowa kluczowe pokrewne: TIME, WHILE

WHILE

WHILE <wyrażenie logiczne>

```
10 CLS:PRINT "Zegar 10 sekundowy":t=TIME
20 WHILE TIME < t+3000
30 SOUND 1,0,100,15
40 WEND:SOUND 129,40,30,15
run
```

ROZKAZ: Powoduje powtarzanie fragmentu programu tak długo, jak długo prawdziwy jest podany warunek.
Słowa kluczowe pokrewne: TIME, WEND

WIDTH

WIDTH <wyrażenie określające liczbę całkowitą>

```
WIDTH 40
```

ROZKAZ: Ustala liczbę znaków w linii w przypadku drukowania na dołączonej drukarce. Po danej liczbie znaków BASIC wysyła dodatkowe znaki sterujące CR i LF (przesunięcie karetki do początku linii i przesunięcie do nowej linii).

Wartość domyślna, zakładana przez komputer, wynosi 132. Rozkaz WIDTH 255 usuwa dodatkowe znaki sterujące CR i LF. Uwaga, znaki CR i LF są wysyłane ZAWSZE przez te rozkazy PRINT, które nie są zakończone średnikiem lub przecinkiem.
Słowa kluczowe pokrewne: POS

WINDOW

WINDOW [# wyrażenie określające strumień >,] < lewa >,< prawa>, <góra>, <dół>

```
10 MODE 0:BORDER 0:REM karty testowe
20 INK 0,0:INK 1,25:INK 2,23:INK 3,21
30 INK 4,17:INK 5,6:INK 6,2:INK 7,26
40 PAPER 0:CLS
50 PAPER 1:WINDOW 2,4,1,18:CLS
```

```
 60 PAPER 2:WINDOW 5,7,1,18:CLS
 70 PAPER 3:WINDOW 8,10,1,18:CLS
 80 PAPER 4:WINDOW 11,13,1,18:CLS
 90 PAPER 5:WINDOW 14,16,1,18:CLS
100 PAPER 6:WINDOW 17,18,1,18:CLS
110 PAPER 7:WINDOW 2,19,19,25:CLS
120 GOTO 120
run
```

ROZKAZ: Określa wymiary strumienia tekstu (okna - WINDOW) na ekranie. Wartości parametrów <lewa>, <prawa>, <góra>, <dół> powinny odpowiadać pozycjom znaków w danym trybie organizacji ekranu.

Jeżeli nie jest podane <wyrażenie określające strumień>, przyjmowana jest wartość #0.

Dalsze informacje dotyczące okien można znaleźć w 2.części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: WINDOW SWAP

WINDOW SWAP

WINDOW SWAP <wyrażenie określające strumień>, <wyrażenie określające strumień>

```
 10 MODE 1:INK 1,24:INK 2,9:INK 3,6
 20 WINDOW 21,40,13,25:PAPER 3
 30 WINDOW #1,1,20,1,12:PAPER #1,2
 40 CLS:PRINT"okno nr.0"
 50 CLS #1:PRINT #1,"okno nr.1"
 60 LOCATE 1,6
 70 PRINT "okno czerwone (0)";SPC (2)
 80 LOCATE #1,1,6
 90 PRINT "okno zielone (1)"
100 FOR t=1 TO 1000:NEXT
110 WINDOW SWAP 0,1:GOTO 60
```

ROZKAZ: Zamienia okno tekstowe określone pierwszym<wyrażeniem> z oknem określonym drugim <wyrażeniem>.

Obydwa <wyrażenia określające strumień> muszą być podane i w tym wypadku, nie powinny być poprzedzone wskaźnikiem kierunkowym .

Rozkaz ten może być używany w celu skierowania komunikatów BASIC'a, które są normalnie kierowane do strumienia # 0, do wybranego okna.

Dalsze informacje dotyczące okien można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili ..."

Słowa kluczowe pokrewne: WINDOW

WRITE

WRITE [#<wyrażenie określające strumień>,][<lista pozycji do wpisu>]

```
10 REM zapisywanie zmiennych na dysku
20 INPUT "podaj zmienną liczbową"; a
30 INPUT "podaj zmienną tekstową"; a$
40 OPENOUT "datafile"
50 WRITE #9,a,a$
60 CLOSEOUT:PRINT "Dane zostały zapisane na dysku"
run
```

ROZKAZ: Zapisuje wartości pozycji z listy w danym strumieniu. Poszczególne pozycje powinny być oddzielone przecinkami; teksty powinny być zamknięte w podwójnych cudzysłowach.

W przykładzie wprowadzone wartości zmiennych są zapisywane w strumieniu #9 (strumień dysku).

W celu odzyskania wartości zmiennych z dysku można użyć następującego programu

```
10 REM odczytanie zmiennych z dysku
20 OPENIN "datafile":INPUT # 9,a,a$
30 CLOSEIN:PRINT "Odczytano 2 wartości:"
40 PRINT:PRINT a,a$
run
```

Słowa kluczowe pokrewne: INPUT, LINE INPUT

XOR

<argument> XOR <argument>

```
IF "piotr"< "zenek" XOR "pies" >"kot" THEN PRINT "dobrze"
ELSE PRINT "źle"
źle
```

```
IF "zenek"<"piotr" XOR "kot" >"pies" THEN PRINT "dobrze"
ELSE PRINT "źle"
źle
IF "piotr"<"zenek" XOR "kot"> "pies" THEN PRINT "dobrze"
ELSE PRINT "źle"
dobrze
...
PRINT 1 XOR 1
0
PRINT 0 XOR 0
0
PRINT 1 XOR 0
1
```

OPERATOR: Wykonuje operację logiczną, wyłącznie - lub, na bitach liczb całkowitych. Bit wyniku jest równy 1, jeżeli bity argumentów są różne.

Dalsze informacje dotyczące logiki można znaleźć w 2. części rozdziału zatytułowanego "W wolnej chwili...".

Słowa kluczowe pokrewne: AND, OR, NOT

XPOS

XPOS

```
10 MODE 1:DRAW 320,200
20 PRINT "pozycja x kursora graficznego=";
20 PRINT XPOS
run
```

FUNKCJA: Podaje aktualną pozycję poziomą (X) kursora graficznego.

Słowa kluczowe pokrewne: MOVE, MOVER, ORIGIN, YPOS

YPOS

YPOS

```
10 MODE 1:DRAW 320,200
20 PRINT "pozycja Y kursora graficznego =";
30 PRINT YPOS
run
```

FUNKCJA: Podaje aktualną pozycję pionową (Y) kursora graficznego.

Słowa kluczowe pokrewne: MOVE, MOVER, ORIGIN, XPOS

ZONE

ZØNE <wyrażenie określające liczbę całkowitą>

```
10 CLS:FOR z=2 TO 20
20 ZONE z
30 PRINT "X","X ZONE=";z:NEXT
run
```

ROZKAZ: Zmienia szerokość strefy drukowania (podawanej w rozkazie PRINT przez użycie przecinka). Wstępnie zakładana szerokość strefy jest równa 13 kolumn i może być zmieniana wartością <wyrażenia> w zakresie 1 do 255.

Słowa kluczowe pokrewne: PRINT

ROZDZIAŁ 4
DYSKI I KASETY

Część 1: Dyski

Tworzenie roboczych kopii dysków

Ta część rozdziału opisuje, jak tworzyć dyski przeznaczone do codziennej pracy oraz wprowadza w pewne możliwości systemu CP/M i współpracujących z nim programów użytkowych.

W szczególności omówione będzie:
- tworzenie kopii dysków systemowych
- wprowadzenie do CP/M PLUS
- Używanie zbiorów HELP
- praca z jednym i dwoma napędami dysków
- kopiowanie zbiorów przy pomocy programu PIP
- operowanie dyskami w formacie języka BASIC
- używanie dysków z programami użytkowymi w języku AMSTRAD BASIC
- dopasowywanie środowiska CP/M PLUS do programu użytkowego
- wprowadzenie do GSX
- praca z systemem CP/M 2.2

W części 7 rozdziału "Informacje podstawowe" znalazły się informacje, jak formatować nową dyskietkę do pracy z językiem BASIC i systemem CP/M.

Z części 10 tego samego rozdziału można dowiedzieć się, jak skopiować dysk, używając programu DISCKIT3 (znajdującego się na 1 stronie pakietu dyskietek systemowych). Ta część podręcznika mówi o tym, jak używać dyski i znajdujące się na nich programy.

Kopiowanie wzorcowych dyskietek systemowych

Utworzenie kopii dostarczonych wraz z komputerem dyskietek systemowych, a następnie trzymanie oryginałów w bezpiecznym miejscu, jest niezwykle ważne. Ich wymiana w wypadku uszkodzenia jest bardzo kosztowna! Pamiętajmy, że każda z wzorcowych dyskietek systemowych zapisana jest po obu stronach, co czyni w sumie cztery strony zapisu. Ogólnie - każdy dysk ma dwie powierzchnie robocze, które mogą być zapisane według życzeń użytkownika.

Pierwsza strona wzorcowego pakietu jest najważniejsza. Zawiera ona wzorcową kopię systemu CP/M PLUS oraz zestaw programów użytkowych, wspomagających operacje dyskowe. Druga strona zapisana jest zbiorami ważnymi dla osób programujących w języku assemblera. Strona trzecia - to Dr. LOGO, zbiory HELP oraz GSX. (O tym ostatnim programie - w dalszej części rozdziału). Czwarta strona zawiera CP/M 2.2 i Dr. LOGO - programy już wcześniej dostępne w modelach AMSTRAD CPC 664 i CPC 464 + DDI1. Programy te dostarczane są z uwagi na kompatybilność z wcześniejszymi modelami Amstradów - normalnie nie są one wykorzystywane.

Kopie dysków systemowych winny być traktowane jako "biblioteki programów". Zwykle program wybierany jest przez wprowadzenie do komputera "dysku bibliotecznego", na który znajduje się poszukiwany zbiór, a nie przez kopiowanie go na pusty dysk i uruchamianie tego ostatniego.

Jeszcze raz należy z naciskiem podkreślić, że "dyski biblioteczne", które używa się w codziennej pracy MUSZĄ być kopiami oryginalnych dysków systemowych.

Pamiętajmy, że przy kopiowaniu zbiorów na pusty dysk przy pomocy programu DISCKIT3, czysty dysk zostaje automatycznie sformatowany.

Wprowadzenie do CP/M PLUS

Zaraz po włączeniu komputera do sieci, na ekranie zgłasza się język BASIC. BASIC zarządzać będzie działaniem systemu tak długo, dopóki nie zostanie wyłączony albo programem BINarnym uruchomionym przez AMSDOS (lub załadowanym z kasety), albo przez załadowanie systemu CP/M PLUS komendą

```
|CPM
```

Po załadowaniu do pamięci komputera, system operacyjny CP/M PLUS nie ma już potrzeby odwoływania się do pierwszej strony dysku systemowego, o ile oczywiście nie pragniemy uruchomić programu użytkowego rezydującego na tej stronie. Tylko dysk, od którego rozpoczynamy pracę z CP/M musi mieć format systemowy; inne dyski mogą już mieć format dający większą pojemność - format danych.

Uruchamianie programu polega na włożeniu dysku zawierającego ten program do napędu dysków i wprowadzeniu z klawiatury odpowiedniej nazwy. Dane używane przez program mogą znajdować się na tym samym bądź innym dysku. CP/M PLUS, podobnie jak AMSDOS, zezwala na wymienianie dysków. Jeżeli z uwagi na wygodę na dysku winien się znajdować pakiet programów, w tym pewne programy uruchomieniowe, radzimy użyć programu PIP ze strony 1 dysku systemowego w sposób opisany w dalszej części tego rozdziału i w rozdziale 5.

Dysk systemowy zawiera specjalny zbiór nazwany PROFILE.SUB, na który składa się lista komend automatycznie wykonywanych po załadowaniu systemu CP/M PLUS do pamięci. Można (o ile to nie zostało jeszcze zrobione) wprowadzić do napędu KOPIĘ pierwszej strony dysku systemowego i po znakach A> wprowadzić

```
REN PROFILE.SUB = PROFILE.ENG
```

co spowoduje utworzenie ze zbioru PROFILE.ENG zbioru PROFILE.SUB. Zbiór ten, który aktywowany będzie już przy następnym załadowaniu CP/M PLUS, zawiera komendy:

```
SETKEYS KEYS.CCP
LANGUAGE 3
```

Powodują one dogodne przy wprowadzaniu komend CP/M PLUS, nadanie znaczenia klawiszom sterowania kursora oraz wybranie zestawu znaków angielskich (a nie amerykańskich). Sprowadza się to do zamiany znaku generowanego przy naciśnięciu [SHIFT]3 z # na £

Od momentu ustawienia znaczenia klawiszy komendą SETKEYS KEYS.CCP, edycja linii komend CP/M może być przeprowadzana analogicznie do edycji linijki w języku BASIC. Dokładny opis działania SETKEYS znajduje się w drugiej części rozdziału 5.

Pomocna dłoń

Trzecia strona pakietu dysków systemowych zawiera specjalny program nazwany "Help". Jest on rodzajem elektronicznego podręcznika programów użytkowych systemu CP/M PLUS. Programu "Help" można użyć wprowadzając do napędu trzecią stronę pakietu i w odpowiedzi na znak gotowości A> wprowadzić

HELP

... a program zada szereg pytań naprowadzających na poszukiwaną informację.

Ile napędów?

Przy pierwszym ładowaniu systemu CP/M automatycznie rozpoznawana jest liczba dołączonych do komputera napędów dysków. Liczba ta wyświetlana jest w nagłówku systemu. Należy zwrócić uwagę, że proces rozpoznawania może być zakłócony, jeżeli dysk w drugim napędzie nie jest całkowicie usunięty.

Wszystkie komunikaty o błędach związanych z napędami wyświetlane są w 25 linii ekranu. Same programy używają tylko pierwszych 24 linii.

Jeżeli dysponujemy tylko jednym, wbudowanym napędem dysków, to ostatnia linia ekranu zawierać będzie tekst "Drive is A:" lub "Drive is B:". Dzieje się tak dlatego, że CP/M PLUS zezwala na pracę z jednym fizycznym napędem tak, jak gdyby do systemu dołączone były dwie logiczne jednostki. Można używać 2 dysków, a ostatnia linia ekranu wyświetlać będzie komunikaty o konieczności zmiany dysku, zgodnie z wymogami programu. Taki tryb pracy pozwala na uniknięcie zakupu drugiego napędu; niesie on jednak ze sobą konieczność manipulowania dyskami i - co za tym idzie - niebezpieczeństwo pomyłek.

Kopiowanie zbiorów z dysku na dysk

W celu udostępnienia możliwości kopiowania zbiorów z dysku na dysk, do zestawu programów użytkowych dołączony jest program PIP ("Peripherial Interchange Program").

Wpierw należy załadować PIP ze strony 1. Dokonuje się tego pisząc po znakach A>

PIP

Po prawidłowym załadowaniu PIP zgłasza się symbolem gwiazdki (*). Zwykle kopiowanie odbywać się będzie z dysku źródłowego (ang. "source") w napędzie A: na dysk docelowy (ang.

"destination") w napędzie B:. Przekonaliśmy się już, że wobec istnienia jednego napędu A: i B: są tym samym mechanizmem.

Aby dokonać kopiowania zbioru, dla przykładu SUBMIT.COM, należy wprowadzić z klawiatury po znaku *

B:=A:SIBMIT.COM

W celu skopiowania wszystkich zbiorów z dysku źródłowego na docelowy należy użyć komendy

B:= *.*

Wyjście z programu PIP i powrót do systemu dokonuje się poprzez wciśnięcie klawisza [RETURN] po znaku *

PIP jest programem bardzo rozbudowanym, o dużych możliwościach. Dalsze szczegóły mogą być znalezione w rozdziale 5.

Dyski tylko z programami w języku BASIC

Jak już wspomniano, dyski o formacie systemowym służą normalnie tylko do inicjacji systemu CP/M PLUS. Dyski używane do zapisywania programów w języku BASIC mogą mieć zatem format danych (tj. bez ścieżek systemowych), zapewniający większość pojemność.

Dysk winien być sformatowany przy pomocy programu DISCKIT3. W celu skopiowania programów na taki dysk należy użyć programu PIP, bądź załadować je do pamięci komendą LOAD a następnie zapisać na dysk komendą SAVE języka BASIC.

Dyski "pod klucz" - AMSTRAD BASIC

Jeżeli kupisz jakieś programy napisane w języku AMSTRAD BASIC na komputer CPC 6128, winny one działać zaraz po włączeniu komputera do sieci, bez dodatkowych zmian w systemie. (Wyrażenie "pod klucz" pochodzi z czasów, gdy wszystkie małe komputery włączało się przy pomocy kluczyka, podobnego do kluczyków samochodowych). Podobnie jak ma to miejsce z dyskami systemowymi - należy pracować używając kopii zakupionego oprogramowania, a oryginały zostawić w bezpiecznym miejscu.

Dyski "pod klucz" - CP/M

System operacyjny CP/M pozwala na korzystanie z olbrzymiej biblioteki programów napisanych na komputery działające pod nadzorem CP/M. Podstawy "logiki" tych programów zostały już przedstawione, wszystko, co należy uczynić w celu ich uruchomienia, to przeniesienie ich na odpowiedni dysk i być może przekazanie im informacji o sposobie, w jaki 6128 obsługuje ekran monitora.

Zestaw programów na dysku, napisanych tak, aby razem wypełniały odpowiednie zadanie, nazywamy "pakietem". Pakiety takie pisane są zwykle tak, aby działały na różnych komputerach, spośród których każdy może mieć inny rozmiar ekranu i sposób sterowania kursora.

Komputer 6128 wyposażony jest we wbudowany "emulator terminala", uruchamiany w czasie ładowania CP/M PLUS. Jego kody kontrolne są różne od używanych przez BASIC.

Czasami pakiet programów użytkowych będzie już "zainstalowany" na AMSTRADA 6128; w innych wypadkach może być dostosowany do 6128 w ten sposób, że dopuszcza instalację kompatybilną z 6128. W takim wypadku należy dokonać wskazanych przez oprogramowanie czynności dla terminala Zenith Z19/Z29. Jeżeli w pakiecie nie ma takich możliwości, to komendy, które po wysłaniu na ekran mogą dopasować 6128 do wymagań oprogramowania, wskazane są w części zatytułowanej "Rekonfiguracja programu pod systemem CP/M". Zwykle proces instalacji lub dostosowania programu sprowadza się do wprowadzenia na żądanie odpowiednich kodów. I tu znowu należy trzymać się instrukcji dołączonej do pakietu.

Zakupione oprogramowanie musi być umieszczone na dysku dostosowanym do użycia w systemie. Niemal każdy komputer używa innego formatu dysku. Mimo, iż wiele ma tę samą średnicę, nic oznacza to automatycznie jakiejkolwiek kompatybilności między nimi w zakresie informacji zawartych na dysku. Należy szukać 3-calowej wersji na komputer AMSTRAD.

Tworzenie "dysku pod klucz" w systemie CP/M

Obok samego programu aplikacyjnego często wygodnie jest mieć na dysku program użytkowy SETKEYS.COM i być może SUBMIT.COM (wraz z odpowiednimi zbiorami zawierającymi komendy).

Do przenoszenia zbiorów typu COM, jak również do pisania zbioru komend wygodnie jest użyć programu PIP. Jeżeli chodzi o tę drugą możliwość, to PIP działa jak edytor poszczególnych linijek tekstu. Dla przykładu, zbiór LOGO3.SUB na stronie 3 pakietu systemowego mógł być utworzony za pomocą następujących komend:

(Należy włożyć dysk systemowy, strona 1 do napędu A:).

Wprowadzamy:

PIP

(Należy wyjąć dysk systemowy, włożyć dysk docelowy).

Wprowadzamy:

```
LOGO3.SUB=CON:
SETKEYS KEYS.DRL
[CONTROL]J LOGO3
[CONTROL]Z
```

Rekonfiguracja programu pod systemem CP/M

CPC 6128 akceptuje szeroki zestaw kodów kontrolnych wygodnych przy rekonfiguracji pakietu tak, aby działał pod CP/M na CPC 6128. Większość procesorów tekstów i innych programów aplikacyjnych jest w stanie wysyłać komunikaty w różne miejsca ekranu i przyjmować teksty pojawiające się w dowolnych jego miejscach oraz ogólnie reagować na kody sterowania kursorem.

Jeżeli Twój pakiet został już odpowiednio dopasowany, dalsze informacje są już zbędne.

Dostosowywanie sposobu wyprowadzania informacji

Dopasowywanie zwykle polegać będzie na wykonaniu specjalnego programu (często nazywanego "INSTAL"). Jeżeli nie uwzględnia on terminali typu Z19/Z29, albo CPC 6128 w szczególności, to należy odpowiedzieć na szereg pytań dotyczących parametrów ekranu 6128. Potencjalne odpowiedzi zebrane są w poniższej tabeli, stanowiącej część informacji z części 15 rozdziału 7.

ASCII	HEX	DEC	Operacja
[BEL]	&07	7	sygnał dźwiękowy
[BS]	&08	8	kursor w lewo o jedną pozycję
[LF]	&0A	10	kursor w dół o jedną linię
[CR]	&0D	13	kursor w lewo do skraju okna w tej samej linii
[ESC]A	&1B &41	27 65	kursor w górę o jedną linię
[ESC]C	&1B &43	27 67	kursor w prawo o jedną pozycję
[ESC]E	&1B &45	27 69	wyczyść ekran
[ESC]H	&1B &48	27 72	kursor do lewego górnego rogu ekranu
[ESC]J	&1B &4A	27 74	wyczyść od znaku pod kursorem do końca ekranu
[ESC]K	&1B &4B	27 75	wyczyść od znaku pod kursorem do końca linii
[ESC]L	&1B &4C	27 76	dodaj linię
[ESC]M	&1B &4D	27 77	usuń linię
[ESC]N	&1B &4E	27 78	usuń znak pod kursorem
[ESC]Y	&1B &59	27 89	ustaw kursor w danej pozycji
	<c> <r>		<c> - numer kolumny + 32
			<r> - numer wiersza + 32
[ESC]d	&1B &64	27 100	wyczyść od początku ekranu do znaku pod kursorem
[ESC]o	&1B &6F	27 111	wyczyść od początku linii do znaku pod kursorem
[ESC]p	&1B &70	27 112	ustaw tryb wyświetlania negatywowego
[ESC]q	&1B &71	27 113	wyłącz tryb wyświetlania negatywowego

Dostosowywanie sposobu wprowadzania informacji

Programy zawarte w pakiecie będą odwoływały się do klawiatury. Większość klawiszy 6128 generuje standardowe kody, za wyjątkiem klawiszy kursora. Zmiany kodów klawiszy można dokonać przy pomocy programu SETKEYS; od programów oczekuje się jednak takiego działania, aby przyjmowały typowe kody.

Nieszczęśliwie się składa, że różne partie oprogramowania oczekują naciśnięcia innych klawiszy w celu uzyskania tych samych funkcji sterujących. Znaki widoczne, "spacja", [TAB] i [RETURN] są w zasadzie uniwersalne, ale nieporozumienia zaczynają się przy [BS], a im dalej, tym gorzej. Porównajmy dla przykładu kody rozumiane jako "ustaw kursor na początku linii" przez:

CP/M - [CONTROL] B

Dr.LOGO - [CONTROL] A

typowy procesor tekstu - [CONTROL]QS

W CPC 6128 zapewnione są trzy układy klawiatury. Każda konfiguracja może być aktywowana ze zbioru na 1 stronie pakietu systemowego. Dla przykładu

SETKEYS KEYS.CCP

tj. jedna z komend wykonywanych automatycznie przez PROFILE.SUB, ustawi klawiaturę tak, aby pasowała do komend systemu CP/M.

Uruchamianie pakietu "pod klucz"

Zwykle wystarczy wpisać nazwę głównego programu w pakiecie po znakach A> . Dla przykładu, aby uruchomić program PAYROLL.COM, wpiszemy

PAYROLL

Jeżeli jakieś dopasowanie jest niezbędne, to pakiet zawierać będzie zbiór typu .SUB. Przykładem tego jest Dr. LOGO na stronie 3 pakietu systemowego. Program ten uruchamiany jest komendą SUBMIT LOGO3. Zawartość zbioru możemy obejrzeć przez wprowadzenie

TYPE LOGO3.SUB

Zobaczymy:

SETKEYS KEYS.DRL	... co zmienia kody klawiatury
LOGO3	... co wywołuje LOGO
SETKEYS KEYS.CCP	... co przywraca kody CP/M.

Automatyczne uruchamianie programu

Można tak zorganizować dysk, aby program aplikacyjny rozpoczynał działanie automatycznie po uruchomieniu CP/M z dysku systemowego. Należy na końcu zbioru PROFILE.SUB dołączyć nazwę interesującego nas programu (który musi rezydować na tym samym dysku).

Wprowadzenie do GSX

GSX (ang. "Graphics System EXtension") jest programem umożliwiającym systemowi operacyjnemu CP/M wyprowadzanie zarówno tekstu jak i obrazów graficznych. GSX używany jest do rysowania takich

rzeczy jak wykresy słupkowe i kołowe, sporządzania nagłówków czcionkami o różnym kształcie i wielkości.

Program GSX umożliwia korzystanie z ekranu monitora, drukarki oraz plotera.

Sam program nie posiada umiejętności rysowania grafiki, podobnie jak CP/M nie potrafi sam przetwarzać tekstów; w obu wypadkach niezbędny jest specjalizowany program. Przykładowy rysunek sporządzony został przez program "DR Graph" firmy Digital Research. Zadaniem GSX jest dostarczenie standardowych procedur związanych z ekranami, drukarkami i ploterami. Pozwala to na przenoszenie oprogramowania aplikacyjnego z jednego komputera na drugi bez konieczności wprowadzenia większych zmian.

W celu stworzenia dysku dostosowanego do wykonywania programów korzystających z GSX należy skopiować na pusty, sformatowany dysk zbiory GSX.SYS i ASSIGN.SYS z 3 strony pakietu systemowego oraz niezbędne programy obsługi fizycznych urządzeń wyjściowych (wraz z samym programem aplikacyjnym). Można w tym celu użyć programu PIP. Zbiór ASSIGN.SYS zawiera opis do trzech programów obsługi urządzeń, w kolejności odwrotnej do ich wielkości:

```
21a:ddfxlr7 ;   drukarka 7-bitowa typu Epson
11a:ddhp747Ø;   ploter
Ø1a:ddmode2 ;   ekran w trybie 2
Ø2a:ddmode1 ;   ekran w trybie 1
Ø3a:ddmodeØ ;   ekran w trybie Ø
```

Liczby wskazują programowi GSX typ urządzenia obsługiwanego przez każdy program obsługi (Drukarka/Ploter/Ekran). W danym momencie w pamięci może rezydować tylko jeden program obsługi, bowiem programy te ładowane są w to samo miejsce pamięci. Stąd potrzeba uszeregowania programów pod względem ich długości - GSX musi zarezerwować tyle miejsca, aby starczyło na największy z nich

Dostarczane są programy obsługi dla różnych trybów pracy ekranu i standardowych drukarek. Wraz z 6128 dostępne są programy obsługi wymienione w zbiorze DRIVERS.GSX. Należy zapoznać się z tym zbiorem poprzez włożenie do napędu dysków 3 strony pakietu i wprowadzenie po znakach A>

TYPF DRIVERS.GSX

Pozwoli to na wybranie odpowiedniego programu.

Ponieważ większość programów aplikacyjnych zawierać będzie program ładujący GSX, wystarczy po znakach A> wpisać nazwę swojego programu. Jeżeli natomiast używany program nie posiada fragmentów ładującego GSX, to należy skopiować zbiór GENGRAF.COM (ze strony 3) na roboczy dysk mający używać GSX i wpisać

GENGRAF YOURFILF

gdzie YOURFILF.COM jest właśnie programem aplikacyjnym, w którym należy zainstalować program ładujący. Skoro od tej chwili program aplikacyjny zawiera już fragment ładujący - zbiór GFNGRAF.COM może już zostać wymazany komendą ERA.

Praca z CP/M 2.2

W przeciwieństwie do CP/M Plus, CP/M 2.2 (tj. starsza wersja systemu CP/M) często odwołuje się do dysku systemowego i nie toleruje zmian dysku w czasie pracy. Wynika to z konieczności powtórnego załadowania fragmentów systemu z dysku w napędzie A: w czasie, gdy po skończeniu działania programu na ekranie pojawiają się znaki A>. Zwykle zatem pracuje się z dyskiem systemowym. W celu przenoszenia zbiorów, w miejsce programu PIP można użyć, nie wymagającego dwóch napędów, programu kopiującego FILFCOPY.

Wobec powyższego, zaleca się używanie CP/M 2.2 jedynie wobec posiadania oprogramowania napisanego na komputerach CPC 464+DDI1 lub CPC 664, z pewnych powodów nie działającego pod CP/M Plus.

Ostrzeżenie: Pewne programy na CPC 664 lub CPC 464 + DDI1 pod CP/M 2.2 posiadają w sobie pewne komendy instalacyjne i nie będą działać pod CP/M Plus. W tych wypadkach z konieczności należy skorzystać z CP/M 2.2, zawartego na 4 stronie pakietu systemowego.

Część 2: Kasety

Jeżeli życzysz sobie używać magnetofonu dołączonego do systemu w sposób opisany w części 2 rozdziału "Kurs podstawowy", to należy pamiętać, że pewna liczba komend języka BASIC, po komendzie |TAPE, zmieni swoje działanie. Na ekranie pojawiać się będą różne komunikaty i znaki, inne niż te, które widoczne są w czasie normalnej pracy z dyskami.

Uwaga. Jeżeli Twój magnetofon ma zdalne sterowanie funkcjami przewijania do tyłu i do przodu, to konieczne będzie odłączenie go, lub użycie komendy |TAPE : CAT w celu uruchomienia silnika. Operacja ta przerywana jest klawiszem [ESC].

W przeciwieństwie do dysków, nazwy zbiorów zapisywanych na magnetofonie nie są obarczone tak ścisłymi ograniczeniami. Mogą one mieć długość do 16 znaków, w tym wtrąconych spacji i znaków przestankowych. W pewnych wypadkach nazwy mogą być równie dobrze ominięte.

Następująca lista opisuje r ó ż n i c e w działaniu poszczególnych komend języka BASIC. Pełen opis koment można znaleźć w rozdziale "Kompletna lista ..."

CAT

Po wprowadzeniu rozkazu CAT wyświetlone zostanie polecenie

Press PLAY then any key :

("Naciśnij klawisz magnetofonu ODTWARZANIE, a następnie dowolny klawisz klawiatury")
co sprowadza się do tego, iż należy włączyć taśmę i zasygnalizować to komputerowi wciskając dowolny klawisz. Taśma w magnetofonie zacznie się kręcić a komputer zacznie wyświetlać nazwy kolejno znalezionych zbiorów na kasecie.

Zostanie wyświetlony każdy blok zbioru, a za nim jeden znak wskazujący typ bloku

$ - to zbiór w języku BASIC, nie zabezpieczony
% - to zbiór w języku BASIC, zabezpieczony
* - to zbiór tekstowy (ASCII)
& - to zbiór binarny

Na końcu każdej linii komputer wyświetli OK, jeżeli zbiór został przeczytany bez błędów; oznacza to, że ładowanie zbioru do pamięci zakończyłoby się powodzeniem.

Funkcja CAT nie narusza zawartości pamięci komputera.

Jeżeli zbiór na kasecie nie został zaopatrzony w nazwę, to CAT wyświetli:

Unnamed file

("Zbiór bez nazwy")

Działanie CAT można przerwać klawiszem [ESC].

Błędy odczytu

Jeżeli żaden z powyższych komunikatów nie pojawi się, bądź otrzymamy na ekranie napis

Read error a ... lub Read error b

to może to oznaczać, że

1) Magnetofon nie jest właściwie podłączony do gniazda TAPE komputera (patrz część 2 rozdziału "Kurs podstawowy")
2) Poziom wyjściowy sygnału z magnetofonu nie jest dopasowany - należy użyć pokręteł GŁOŚNOŚĆ lub POZIOM
3) Jakość taśmy jest słaba, taśma ma zagięcia
4) Taśma uległa rozmagnesowaniu w czasie znajdowania się w bliskim sąsiedztwie głośników lub telewizora
5) Taśma nie jest przeznaczona do odczytu przez komputery AMSTRAD

CHAIN
CHAIN MERGE
LOAD
MERGE
RUN

Jeżeli pragniesz załadować z taśmy pierwszy napotkany zbiór, nie jest konieczne określanie jego nazwy.

Przykładowe komendy:

```
CHAIN " "
CHAIN " ", 100

CHAIN MERGE " ",
CHAIN MERGE " ", 100
CHAIN MERGE " ", 100, DELETE 30-70

LOAD " "
LOAD " ", & 1F40

MERGE " "

RUN " "
```

Należy zwrócić uwagę, że ostatnia komenda wykonuje się samoczynnie, gdy naciskamy klawisz [ENTER] trzymając jednocześnie klawisz [CONTROL]. Należy używać tego sposobu korzystając z oprogramowania na kasecie, po komendzie |TAPE

Komputer poinstruuje;

Press PLAY then any key

/co oznacza: naciśnij klawisz magentofonu PLAY a następnie dowolny klawisz komputera/

Taśma zacznie się obracać, a komputer załaduje odpowiedni zbiór.

Na ekranie ukaże się następujący komunikat:

Loading FILENAME block 1

/co oznacza: ładowany jest zbiór FILENAME, blok 1/ i będzie to powtarzane tyle razy, ile bloków zawiera ładowany program.

Jeżeli pierwszym znakiem zbioru jest wykrzyknik / ! /, to powyższy komunikat nie będzie wyświetlany; nie będzie też konieczne naciskanie żadnych klawiszy. /Należy zatem pamiętać o uruchomieniu magentofonu!/. Jeżeli takie programy są również uruchamiane z dysku, to wykrzyknik zostanie w czasie czytania nazwy zbioru z dysku pominięty. Zwróćmy uwagę, że ! nie zajmuje żadnej pozycji w nazwie zbioru - ani zapisanego na kasecie, ani na dysku.

Przerwanie operacji poprzez naciśnięcie ESC powoduje wyprowadzenie na ekranie komunikatu

Broken in

Jeżeli ładowanie zbioru nie zakończyło się powodzeniem - wróć do fragmentu tego rozdziału pod tytułem "Błędy odczytu".

Ostrzeżenie

Wewnętrzny interfejs dysku zajmuje pewien obszar pamięci; obszar ten był używany przez niektórych zawodowych programistów przy okazji tworzenia oprogramowania na kasetach. Takie programy nie będą działać właściwie na CPC 6128 z dołączonym magnetofonem.

EOF

POS(#9)

Funkcje te działają na kasetę podobnie jak na dysk

INPUT #9

LINE INPUT #9

OPENIN i CLOSEIN

Jeżeli użyty ma być pierwszy zbiór na taśmie, to nie trzeba podawać nazwy zbioru. Przykładowa komenda:

OPENIN" "

Wyświetlona będzie instrukcja:

Press PLAY then any key

... co oznacza, że winniśmy nacisnąć klawisz PLAY magnetofonu, a następnie dowolny klawisz komputera. Taśma zacznie się kręcić, a komputer załaduje do umiejscowionego w pamięci "bufora zbioru" pierwsze 2KB zbioru. Następnie dane z bufora są pobierane aż do jego opróżnienia, po czym znowu wyświetlone zostanie polecenie

Press PLAY then any key

Na ekranie pojawi się napis

Loading FILENAME block 1

i, w czasie ładowania kolejnych bloków, kolejne informacje

Jeżeli pierwszym znakiem zbioru w komendzie OPENIN jest wykrzyknik (!), to powyższy komunikat nie będzie wyświetlany; nie będzie też konieczne naciskanie żadnych klawiszy. (Należy zatem pamiętać o uruchomieniu magnetofonu !). Jeżeli takie programy są również uruchamiane z dysku, to wykrzyknik zostanie w czasie czytania nazwy zbioru z dysku pominięty. Zwróćmy uwagę, że ! nie zajmuje żadnej pozycji w nazwie zbioru - ani zapisanego na kasecie, ani na dysku.

Przerwanie operacji poprzez naciśnięcie ESC powoduje wyprowadzenie na ekranie komunikatu

```
Broken in
```

Jeżeli ładowanie zbioru nie zakończyło się powodzeniem - wróć do fragmentu tego rozdziału pod tytułem "Błędy odczytu".

LIST #9
OPENOUT i CLOSEOUT
PRINT #9
WRITE #9

Jeżeli pragniemy zapisać zbiór jako "Unnamed-file" (tj. "Zbiór bez nazwy"), to nie musimy podawać nazwy zbioru. Przykładowa komenda:

```
OPENOUT" "
```

Pierwsze 2KB zbioru zostanie wpierw przeniesione do obszaru pamięci nazywanego "buforem zbioru". Gdy bufor będzie już pełen, wyświetlone zostanie polecenie:

```
Press REC and PLAY then any key
```

... co oznacza, że powinniśmy wcisnąć w magnetofonie klawisze RECORD ("Nagrywanie") i PLAY, a następnie nacisnąć dowolny klawisz komputera. Taśma zacznie się obracać, a komputer zapisze na nią zawartość bufora. Następnie komputer powtórnie zapełni bufor kolejnymi 2KB zbioru i wyświetli po raz drugi

```
Press REC and PLAY then any key
```

... zapisując po chwili bufor na taśmę.

Jeżeli bufor zbioru jest częściowo tylko zapełniony i komputer napotkał komendę CLOSEOUT, pozostała zawartość zbioru zostanie zapisana na taśmę i pojawi się

Press REC and PLAY then any key

Na ekranie zostanie wyświetlony napis

Saving FILENAME block <x>

Jeżeli pierwszym znakiem zbioru w komendzie OPENOUT jest wykrzyknik (!), to powyższy komunikat nie będzie wyświetlany; nie będzie też konieczne naciskanie żadnych klawiszy. (Należy zatem pamiętać o uruchomieniu magnetofonu!). Jeżeli takie programy są również uruchamiane z dysku, to wykrzyknik zostanie w czasie czytania nazwy zbioru z dysku pominięty. Zwróćmy uwagę, że ! nie zajmuje żadnej pozycji w nazwie zbioru - ani zapisanego na kasecie, ani na dysku.

Przerwanie operacji poprzez naciśnięcie ESC powoduje wyprowadzenie na ekranie komunikatu

Broken in

Udany zapis na taśmę

W celu zapewnienia prawidłowego zapisu:

1) Sprawdź, czy magnetofon jest prawidłowo połączony z gniazdem TAPE komputera (patrz część 2 rozdziału "Wiadomości podstawowe")
2) Sprawdź, czy pokrętła POZIOM i ZAPIS w magnetofonie są prawidłowo ustawione
3) Sprawdź, czy nie zamierzasz użyć kasety słabej jakości lub C120 (Polecane są kasety AMSOFT C15)
4) Sprawdź, czy magnetofon nie jest wystawiony na działanie pól magnetycznych (działające w pobliżu głośniki lub telewizor)
5) Przed usunięciem programu z pamięci komputera, sprawdź skorowidz (CAT) taśmy - upewnij się, czy program jest prawidłowo nagrany
6) Zapewnij okresową konserwację magnetofonu i regularne czyszczenie głowic

SAVE

Jeżeli nie nadana zostanie zbiorowi żadna konkretna nazwa, to zostanie on zapisany jako "Unnamed file" ("Zbiór bez nazwy"). Przykładowa komenda:

SAVE " "

Komputer wyświetli polecenie:

Press REC and PLAY then any key

po czym należy wcisnąć w magnetofonie klawisz RECORD ("Nagrywanie") i PLAY, a następnie dowolny klawisz komputera. Taśma zacznie się kręcić, a komputer zapisze program.

Na ekranie pojawi się napis

Saving FILENAME block 1

a potem kolejno napisy wskazujące na zapisywanie kolejnych bloków, aż do końca programu.

Jeżeli pierwszym znakiem zbioru jest wykrzyknik (!), to powyższy komunikat nie będzie wyświetlany; nie będzie też konieczne naciskanie żadnych klawiszy. (Należy zatem pamiętać o uruchomieniu magnetofonu !). Jeżeli takie programy są również uruchamiane z dysku, to wykrzyknik zostanie w czasie czytania nazwy zbioru z dysku pominięty. Zwróćmy uwagę, że ! nie zajmuje żadnej pozycji w nazwie zbioru - ani zapisanego na kasecie, ani na dysku.

Przerwanie operacji poprzez naciśnięcie ESC powoduje wyprowadzenie na ekranie komunikatu

Broken in

Prosimy przeczytać wcześniejszy akapit "Udany zapis".

SPEED WRITE

("Szybki zapis"). Ta komenda dotyczy wyłącznie zapisu na magnetofon i może być wydana, gdy komputer ustawiony jest na pracę z dyskiem.

Komunikaty o błędach

W czasie operacji na taśmie mogą być wygenerowane komunikaty o błędach o numerach 7, 21, 24, 25, 27 i 32. (Patrz - część 6 rozdziału zatytułowanego "Trochę użytecznych informacji").

Komendy zewnętrzne systemu AMSDOS

Przełączanie strumieni wejścia/wyjścia między dyskiem i taśmą dokonuje się za pomocą komend

|TAPE (która dzieli się na |TAPE.IN i |TAPE.OUT

|DISC (która dzieli się na |DISC.IN i |DISC.OUT

Zestawienie pozostałych komend zewnętrznych

|A

|B

|CPM

|DIR

|DRIVE

|ERA

|REN

|USER

Są to komendy działające tylko na dysk, niezależnie od tego, czy aktualnie pracuje się z dyskiem czy taśmą.

ROZDZIAŁ 5

AMSDOS i CP/M

Część 1: AMSDOS

Omawiane zagadnienia:
- Co to jest AMSDOS
- Katalog dysku
- Zmiana dysków
- Nazwy zbiorów i typy zbiorów
- Automatyczne określenie typów zbiorów
- Nazwy zbiorów w systemie z dwoma dyskami
- Znaki zastępcze
- Przykładowy program z rozkazami AMSDOS
- Wykaz rozkazów AMSDOS
- Kopiowanie i manipulowanie zbiorami
- Sygnalizowanie błędów

Wstęp

AMSDOS rozszerza język AMSTRAD BASIC, stosowany w Twoim komputerze o szereg dodatkowych rozkazów, wyróżnianych za pomocą poprzedzającego je znaku | (pionowej kreski).

AMSDOS umożliwia użytkownikowi swobodną wymianę dysków, jeżeli tylko nie jest prowadzona operacja odczytu lub zapisu zbiorów; zmiana dysku w czasie trwania operacji odczytu lub zapisu powoduje wyprowadzenie informacji o błędzie i może prowadzić do straty danych, wpisywanych do otwartego zbioru

Katalog /skorowidz/ dysku

Każdy dysk w specjalnie wydzielonej części zawiera katalog /skorowidz/ zapisanych na nim zbiorów (danych). W skorowidzu umieszczone są nazwy wszystkich zbiorów i "mapa" określająca rozmieszczenie ich na dysku. Na podstawie danych, umieszczonych w skorowidzu AMSDOS lub CP/M może określić rozmiar każdego ze

zbiorów oraz wolne miejsce, pozostałe na dysku do dalszego wykorzystania.

Przy odczycie zbioru przeglądany jest skorowidz dysku i określane miejsce jego zapisu. Przy zapisie nowego zbioru umieszczany jest on w wolnych do tej pory obszarach dysku a kasowanie zbioru uwalnia zajmowane przez ten zbiór miejsce. Skorowidz opisuje zbiór w jednostkach 1 KB i zawierać może do 64 pozycji. Jedna pozycja skorowidzu opisać może zbiór o objętości do 16 KB, większe zbiory zapisywane są na kilku pozycjach, chociaż nie jest to zauważane przez użytkownika.

Zmiana dysków

AMSDOS (i CP/M Plus) zezwala na wyjmowanie i wymianę dysków w dowolnej chwili, kiedy napędy nie pracują i nie są otwarte żadne zbiory wejściowe lub wyjściowe. Nie jest tu konieczne wstępne "zaksięgowanie" dysku, jakie wymagane jest w systemie CP/M 2.2.

Wymiana dysku w czasie trwania operacji zapisu może zniszczyć dane na dysku. Jeżeli dysk zmieni się w sytuacji, kiedy pozostają na nim otwarte zbiory, to po wykryciu tego przez AMSDOS dalsze operacje z nie zamkniętymi zbiorami zostają zaniechane i wyprowadzana jest informacja o błędzie. Dane wpisane do tych zbiorów do chwili przerwania będą stracone, gdyż nie zostanie wypełniona odpowiednia pozycja skorowidzu dyskowego. Jednakże AMSDOS może wykryć zmianę dysku jedynie podczas czytania skorowidzu, co wykonywane jest co 16 KB zbioru, lub w trakcie otwierania lub zamykania zbioru. Tak więc wymiana dysku z otwartymi zbiorami prowadzić może do zniszczenia do 16 KB danych.

Nazwy i typy zbiorów w systemie AMSDOS

Normalnie praktykuje się wprowadzanie takich nazw zbiorów dyskowych, aby określały one typ zbioru. Nie zmusza to komputera do używania zbiorów w jakikolwiek szczególny sposób, chociaż

niektóre programy akceptują tylko zbiory o ściśle określonym typie. AMSDOS może przyjąć dowolny typ nazwy zbioru, chociaż nastawiony jest na wyszukiwanie zbiorów o pewnych określonych typach, jeżeli nie otrzymał szczegółowych poleceń (patrz "Automatyczne określanie typów zbiorów")

Budowa nazwy zbioru

Nazwa zbioru składa się z dwóch części, rozdzielonych kropką (.). Pierwsza część zawierać może do 8, a druga do 3 znaków. Prawidłowymi nazwami zbiorów są np.: "ROINTIME.DEM", "DISCKIT3.COM" i "DISC.BAS". Obydwie części nazwy zbioru mogą być tworzone przez kombinację liter lub cyfr, nie mogą w nich natomiast występować spacje lub kropki.

Druga część nazwy zbioru jest nazywana typem zbioru. Niektóre powszechnie używane typy zbiorów to:

. <spacja>	Typ nieokreślony. Może to być zbiór danych tworzony przez OPENOUT "<nazwa zbioru>" lub program w języku BASIC zapisywany przez AMSDOS przy użyciu polecenia SAVE "<nazwa zbioru>", A
.BAS	Program w języku BASIC zapisywany przez AMSDOS rozkazem SAVE "<nazwa zbioru>" lub SAVE "<nazwa zbioru>", P albo SAVE "<nazwa zbioru>" .BAS, A
.BIN	Program lub zawartość obszaru pamięci zapisana przez AMSDOS rozkazem SAVE "<nazwa zbioru>", B, <parametry binarne>
.BAK	Poprzednia wersja zbioru, jeżeli AMSDOS lub inny program użytkowy zapisał nową wersję zbioru o tej samej nazwie. Umożliwia to użytkownikowi ewentualny powrót (BAcK) do starej wersji zbioru
.COM	Zbiór /program/ wykorzystywany jako polecenie systemowe (COMmand); wszystkie programy systemowe w systemie CP/M są zapisane w postaci zbiorów tego typu
.SUB	Zbiór poleceń dla programu SUBMIT w systemie CP/M

Automatyczne określanie typów zbiorów

AMSDOS samoczynnie zapisuje (poleceniem SAVE) zbiory o odpowiednio dobranym typie zbioru tak, że nie jest konieczne określanie typu zbioru, chyba że chce się przypisać nazwie zbioru inny typ, niż przydzielany automatycznie zgodnie z zamieszczoną wyżej tabelą. Programy w języku BASIC, programy BASIC zabezpieczone i zbiory binalne zapisywane są na dysk wraz z odpowiednim nagłówkiem i rozkaz typu:

LOAD "<nazwa zbioru>"

... umożliwia rozpoznanie typu zbioru i właściwe jego wprowadzenie do pamięci komputera. Jeżeli po rozkazie LOAD AMSDOS nie znajdzie w nagłówku określenie typu zbioru, to zakłada, że zbiór zawiera program zapisany w ASCII, jawnym tekstem. Po komendzie LOAD "<nazwa zbioru>", w której nie jest określony typ zbioru najpierw poszukiwany jest zbiór typu:

. <spacja>

Jeżeli zbiór taki nie istnieje, poszukiwany jest zbiór typu:

. BAS

... i wreszcie zbiór typu:

. BIN

Umożliwia to użytkownikowi skracać określenie zbioru i w większości przypadków nie wprowadzać typu zbioru.

Dyskowe zbiory danych, otwierane poleceniem OPENOUT i sukcesywnie zapisywane na dysk nie mają przypisanego typu zbioru i powinny zawierać jawny tekst pisany w ASCII, wpisywany do zbioru rozkazami WRITE, PRINT lub LIST.

Rozkaz OPENIN, w którym nie określono typu zbioru, powoduje przeszukiwanie zbiorów w takiej samej kolejności, zakładanych typów, jak rozkład LOAD.

Nazwy zbiorów w systemie z dwoma napędami dyskowymi

W systemie z dwoma napędami dyskowymi, tzn. w przypadku dołączenia do komputera dodatkowego napędu dyskowego, operować można zbiorami występującymi na dwóch dyskach. Komputer nie

poszukuje automatycznie zbiorów na obu dyskach, lecz tylko na jednym, określonym przez użytkownika. W tym celu należy wprowadzać polecenie |A, |B lub |DRIVE (w pełni opisane nieco dalej), aby wybrać jeden z dwu napędów dyskowych, a następnie napisać nazwę zbioru lub poprzedzić nazwę zbioru przedrostkiem A: albo B:. Tak więc, na przykład zarówno:

```
|B
SAVE "PROG.BAS"
|A
```

jak i:

```
|A
SAVE "B:PROG.BAS"
```

powoduje zapisanie programu na dysku, umieszczonym w dodatkowym napędzie B.

Podobnie zmienić można numer użytkownika (USER) przez podanie numeru z zakresu od 0 do 15 jako przedrostka nazwy zbioru. Rozkazy:

```
LOAD "15:PROG.BAS"
```

i

```
SAVE "15:PROG.BAS"
```

ładują i zapisują program dostępny tylko dla użytkownika (USER), oznaczonego numerem 15.

Można także określić jednocześnie numer użytkownika i dysk, wprowadzając odpowiednie oznaczenia przed nazwą zbioru, na przykład:

```
RUN "15B:PROG.BAS"
```

Znaki zastępcze

Często wymagane jest wykonanie określonych operacji (kopiowanie, kasowanie itp.) kolejno na kilku zbiorach. Wprowadzenie nazwy zbioru powoduje wyszukanie w skorowidzu dyskowym zbioru, którego nazwa dokładnie, co do znaku odpowiada nazwie umieszczonej w rozkazie operacji. W rozkazach dotyczących niektórych operacji można jednakże zastąpić wybrane znaki nazwy zbioru

znakiem ?, co umożliwia wykonanie operacji na pewnej grupie zbiorów. Jeżeli cała nazwa lub dalsza część nazwy z wyjątkiem kilku pierwszych znaków zastąpiona jest znakami ? , grupę tych znaków zastąpić można symbolem *. I tak na przykład FRED.* jest równoważnym skrótem FRED. ??? a F*.BAS zastępuje F???????.BAS. I wreszcie wyrażenie *.* oznacza "wszystkie zbiory". Poniżej przedstawiono przykład selekcji zbiorów na podstawie nazw ze znakami zastępczymi:

SKOROWIDZ DYSKU	Zbiory wybierane przez: *.BAS	FRED?.BAS	F*.BA?
BERT.BAS	BERT.BAS		
FRED1.BAS	FRED1.BAS	FRED1.BAS	FRED1.BAS
FRED2.BAS	FRED2.BAS	FRED2.BAS	FRED2.BAS
FRED3.BAK			FRED3.BAK
FRED3.BAS	FRED3.BAS	FRED3.BAS	FRED3.BAS
FINISH.BAS	FINISH.BAS		FINISH.BAS

Przykłady użycia w programie dyrektyw systemu AMSDOS

Dla lepszego zrozumienia dyrektyw systemu AMSDOS, proponujemy prześledzenie poniższych przykładów, korzystając przy tym z wyjaśnień zawartych w dalszej części tego rozdziału. NIE WPISUJ i NIE URUCHAMIAJ tych programów, jeżeli w komputerze umieszczony jest któryś z oryginalnych dysków z systemem CP/M.

Przechowanie zmiennych i przechowanie zawartości ekranu

Przedstawiany program zapisuje dane na dysku; przed jego uruchomieniem należy zatem umieścić czysty (sformatowany) lub roboczy dysk w napędzie dyskowym. Program rysuje na ekranie flagę angielską ("Union Jack") a następnie wpisuje zawartość całego ekranu na dysk.

```
10 dumpfile$="flagdump.srn"
20 MODE 1:BORDER 0
30 DIM colour(2)
40 FOR i=00 TO 2
50 READ colour(i):REM wez kolory z DATA
60 INK i,coloour(i)
70 NEXT
80 ON ERROR GOTO 430
90 OPENIN "param.dat" 'sprawdz czy zbior istnieje
100 CLOSEIN:ON ERROR GOTO 0
110 IF errnum=32 AND DERR=146 THEN CLS:
    GOTO 160 'zbior nie istnieje
120 CURSOR 1:PRINT "Czy chcesz zmienic stary zbior? Y/N";
130 a$=INKEY$:ON INSTR(" YN",UPPER$(a$))
    GOTO 130,150,140:GOTO 130
140 PRINT a$:PRINT "Program przerwany":END
150 PRINT a$:CURSOR 0
160 OPENOUT "param.dat"
170 WRITE #99,dumpfile$,1:
    REM przechowaj nazwe zbioru i tryb pracy ekranu
180 FOR i=0 TO 2
190 WRITE #9,colour(i): REM przechowaj kolory
200 NEXT i
210 CLOSEOUT
220 CLS
230 gp=1:GRAPHICS PEN gp:w=125
240 x=-65:a=240:y=400:b=-150:GOSUB 400
250 y=0:b=150:GOSUB 400
260 x=575:a=-240:y=400:b=-150:GOSUB 400
270 y=0:b=150:GOSUB 400
280 gp=2:GRAPHICS PEN gp:w=40
290 a=240:x=-40:y=400:b=-150:GOSUB 400
300 x=0:y=0:b=150:GOSUB 400
310 a=-240:x=640:y=0:b=150:GOSUB 400
320 x=6000:y=400:b=-150:GOSUB 400
330 ORIGIN 0,0,256,380,0,400:CLG 1
340 ORIGIN 0,0,0,640,150,250:CLG 1
350 ORIGIN 0,0,280,352,0,400:CLG 2
360 ORIGIN 0,0,0,640,168,230:CLG 2
370 SAVE dumpfile$,b,&C000,&4000
380 DATA 2,26,6
390 END
400 MOVE x,y:DRAWR a,b:DRAWR w,0:DRAWR -a,-b
410 MOVE x+a/2+w/2,y+b/2:FILL gp
420 RETURN
430 errnum=ERR:RESUME NEXT
```

Zwróć uwagę na użycie nazw zbiorów typu .DAT i .SRN. Symbole takie nie mają żadnego istotnego znaczenia, ułatwiają jedynie zapamiętanie co przechowuje zbiór (DATA - dane i ScReeN - ekran). Zbiór PARAM.DAT założony zostanie jako zbiór danych tekstowych (ASCII) typu nieokreślonego zaś zbiór FLAGDUMP.SRN jako zbiór binarny.

Zauważ, że program próbuje czytać ze zbioru PARAM.DATA przed zapisem do tego zbioru aby ustalić, czy zbiór taki już istnieje. Jeżeli zbiór NIE istnieje, BASIC zgłasza błąd, co jest przechwytywane przez program w sposób, umożliwiający wykonywanie programu bez jego przerywania. W przypadku gdy zbiór już ISTNIEJE, nie jest zgłaszany żaden błąd i program pyta się, czy chcesz wpisać nową treść zbioru w miejsce dotychczasowej.

Niektóre dane, opisujące stan ekranu jak tryb wyświetlania (sreen mode), kolory i nazwa zbioru, zawierającego aktualne informacje przechowywane są w osobnym zbiorze z parametrami. Ilustruje to użycie zbioru danych, do którego wpisuje się (rozkazem WRITE) programowe zmienne (dumpfile$) i stałe (1) w celu przechowania ich do użycia przez inny program.

Odtworzenie treści ekranu

Poniżej przedstawiono przykład uniwersalnego programu wpisującego treść ekranu, którego działanie kontrolowane jest przez zbiór z parametrami. Proszę zauważyć, jak zmienne pobierane są ze zbioru dyskowego za pomocą rozkazu INPUT oraz użycie funkcji EOF (End Of File - koniec zbioru) do samoczynnej zmiany wielkości zbioru. Należy pamiętać, że treść ekranu, wpisywanego przez ten program została zapisana w ustalonych adresach pamięci a ich zmiana prowadziłaby do zniekształceń obrazu. Dlatego w programie wykonywana jest instrukcja MODE z parametrem, przechowywanym w zbiorze danych. Ponadto trzeba uważać, aby nie spowodować przewijania /skrolowania/ treści ekranu.

```
10 DIM colour(15):REM miejsce dla 16 kolorow
20 OPENIN "param.dat"
30 INPUT #9, filename$,screenmode
40 i=0
50 WHILE NOT EOF
60  INPUT #9,colour(i)
70 INK i,colour(i)
80 i=i+1
90 WEND
100 CLOSEIN
110 MODE screenmode:BORDER 0
120 LOAD filename$
```

Wykaz rozkazów systemu AMSDOS, dotyczących operacji zewnętrznych

|A

|A

ROZKAZ: Ustanawia jako domyślny napęd dyskowy A.
Odpowiednik rozkazu |DRIVE z parametrem A
(Symbolem A oznaczony jest główny napęd dyskowy w komputerze)

|B

|B

ROZKAZ: Ustanawia jako domyślny napęd dyskowy B.
Odpowiednik rozkazu DRIVE z parametrem B.
(Główny napęd dyskowy w komputerze oznaczony jest symbolem A)

|CPM

|CPM

ROZKAZ: Uruchamia system operacyjny CP/M przez wprowadzenie tego systemu z dysku. W oprogramowaniu systemowym komputera dostarczane są dwie odmiany systemu CP/M: CP/M Plus i CP/M 2.2. Do wykonania tego rozkazu konieczne jest umieszczenie w głównym napędzie dyskowym odpowiedniego dysku systemowego. Z kopii Strony 1 ładowany jest CP/M Plus, z kopii Strony 4 ładowany jest starszy system CP/M 2.2.

|DIR

|DIR [,<wyrażenie tekstowe>]

|DIR, "*.BAS"

ROZKAZ: Wyświetla skorowidz dysku w takiej samej formie, jak system CP/M oraz podaje pozostałe na dysku wolne miejsce. Jeżeli <wyrażenie tekstowe> zostanie pominięte, przyjmowane jest zamiast niego wyrażenie zastępcze *.* /wszystkie zbiory/.

|DISC

|DISC

ROZKAZ: Łączy w sobie jednocześnie dwa rozkazy: |DISC.IN i |DISC.OUT

|DISC.IN

|DISC.IN

ROZKAZ: Oznacza użycie dysku jako źródła zbioru wejściowego

|DISC.OUT

|DISC.OUT

ROZKAZ: Oznacza użycie dysku do umiejscowienia zbioru wyjściowego

|DRIVE

|DRIVE,<wyrażenie tekstowe>

|DRIVE, "A"

ROZKAZ: Ustanawia jako domyślny określony napęd dyskowy. Rozkaz nie zostanie efektywnie wykonany jeżeli AMSDOS nie będzie mógł czytać dysku z żądanego napędu dyskowego (nie dołączony drugi napęd lub nie włożony dysk)

|ERA

|ERA,<wyrażenie tekstowe>

|ERA, "*.BAK"

ROZKAZ: Kasuje (ERAses) wszystkie zbiory o podanej nazwie, jeżeli nie są to zbiory "tylko do czytania" (Read/Only). W <wyrażeniu tekstowym> występować mogą znaki zastępcze (? *).

|REN

|REN,<wyrażenie tekstowe>, <wyrażenie tekstowe>

|REN, "NEWNAME.BAS", "OLDNAME.BAS"

ROZKAZ: Nadaje zbiorom nową nazwę (REName). Nie może to być nazwa istniejącego już na dysku zbioru. Nie dopuszcza się stosowania znaków zastępczych (?*).

W <wyrażeniu tekstowym> może być wprowadzony numer użytkownika (USER). Dla przykładu, polecenie:

|REN, "Ø:NEW.BAS","15:OLD.BAS"

zmienia nazwę zbioru OLD.BAS użytkownika nr. 15 na NEW.BAS użytkownika nr. Ø, zmieniając jednocześnie przypisanie zbioru do użytkownika

|TAPE

|TAPE

ROZKAZ: Łączy w sobie jednocześnie dwa rozkazy: |TAPE.IN i |TAPE.OUT. Używany w przypadku dołączenia zewnętrznego magnetofonu.

|TAPE.IN

|TAPE.IN

ROZKAZ: Oznacza użycie taśmy magnetofonowej (TAPE) jako nośnika zbioru wejściowego. Używany w przypadku dołączenia zewnętrznego do magnetofonu

|TAPE.OUT

|TAPE.OUT

ROZKAZ: Oznacza użycie taśmy magnetofonowej (TAPE) jako nośnika zbioru wyjściowego. Konieczne dołączenie zewnętrznego magnetofonu

|USER

|USER, <wyrażenie określające liczbę całkowitą>

|USER,3

ROZKAZ: Nakazuje prowadzenie wszelkich operacji dyskowych (np. CAT, LOAD, |DIR itp.) tylko na zbiorach użytkownika, określonego odpowiednim numerem (od 0 do 15). Jeżeli nie użyto wcześniej tego rozkazu, domyślnie przyjmowany jest użytkownik nr. 0.

Zbiór określonego użytkownika może być przekazany innemu użytkownikowi za pomocą rozkazu |REN. Na przykład, |REN,"15: EXAMPLE.BAS", "Ø:EXAMPLE.BAS" zmienia przyporządkowanie zbioru użytkownika nr. 15 użytkownikowi nr. Ø przy zachowaniu tej samej nazwy zbioru (EXAMPLE.BAS).

Kopiowanie zbiorów z dysku na dysk

Zbiory zakładane przez AMSDOS z określonym typem zbioru

Zbiory takie mogą być kopiowane za pomocą programu PIP w systemie CP/M (patrz część 2 tego rozdziału). Każdy typ zbioru zakładanego przez AMSDOS, opisany wcześniej w punkcie "Automatyczne określanie typów zbiorów" jest kopiowany w całości - z dysku na dysk - chociaż treść tych zbiorów nie jest zrozumiała dla żadnego programu CP/M.

Zbiory ASCII

Zbiory typu nieokreślonego, zakładane przez AMSDOS są zbiorami tekstowymi, zapisywanymi w kodzie ASCII i mogą być kopiowane i używane (rozumiane) przez programy CP/M. W szczególności możliwa jest swobodna wymiana programów zapisanych w ASCII i zbiorów danych zapisanych w ASCII między programami systemu AMSDOS i systemu CP/M.

Zbiory zabezpieczone przed zapisem, "tylko do czytania" (Read/Only)

Jest możliwe przy użyciu systemu CP/M nadanie dowolnemu zbiorowi statusu "zbioru tylko do czytania" lub specjalnego statusu ukazywanie się /lub nie ukazywanie/ w wyświetlanym skorowidzu dysku. Atrybuty takie mogą być przypisywane lub kasowane tylko w systemie CP/M, lecz są one respektowane przez MASDOS. Więce szczegółów podano w 2 części tego rozdziału (dyrektywa SET).

Kopiowanie zbiorów z dysku na taśmę magnetofonową lub odwrotnie

W systemie CP/M Plus nie ma żadnej możliwości dostępu do zbiorów zapisanych na taśmie magnetofonowej ani kopiowania tych zbiorów na dysk /lub odwrotnie/. Jeżeli wymagane jest kopiowanie takich zbiorów, należy użyć system CP/M 2.2 (ze strony 4 dysków systemowych). CP/M 2.2 zawiera programy CLOAD i CSAVE, a ich stosowanie przedstawiono w zamieszczonych dalej tablicach.

Procedury kopiowania zbiorów

W poniższych tablicach ujęto wszystkie przypadki kopiowania zbiorów między dyskiem a taśmą magnetofonową (z dołączonego magnetofonu). Założono przy tym, że nie jest dołączony dodatkowy napęd dyskowy. Nie jest możliwe jakiekolwiek kopiowanie zabezpieczonych programów BASIC na taśmę albo z taśmy oraz kopiowanie zbiorów binarnych (takich jak np. gry zapisane w kodzie maszynowym) na taśmę albo z taśmy, jeżeli nie są znane adresy ładowania. Więcej szczegółów o programach PIP, CLOAD i CSAVE zamieszczono w 2 części tego rozdziału.

Dwie następne strony zawierają tablice kopiowania:

<table>
<tr><th rowspan="2">KOPIOWANIE DO</th><th colspan="3">KOPIOWANIF Z TAŚMY</th></tr>
<tr><th>AMSTRAD BASIC
zbiór typu .BAS</th><th>Zbiór ASCII
typ nieokreślony</th><th>Zbiór binarny
typu .BIN</th></tr>
<tr><td>zbiór AMSTRAD
BASIC typu .BAS
na taśmie</td><td>|TAPF
LOAD "ZBIÓR"
zmienić taśmę
SAVF "ZBIÓR"
|DISC</td><td></td><td></td></tr>
<tr><td>zbiór binarny typu
.BIN na taśmie</td><td></td><td></td><td>H=HIMEM
|TAPF
MFMORY s -1
LOAD "ZBIÓR"
zmienić taśmę
SAVF "ZBIÓR",B,
s , l , r
|DISC
MFMORY H
uwaga 2</td></tr>
<tr><td>zbiór ASCII
na taśmie</td><td>|TAPF
LOAD "ZBIÓR"
zmienić taśmę
SAVE "ZBIÓR",A
|DISC</td><td>włożyć dysk z CP/M 2.2
|CPM
CLOAD "ZBIÓR",TEMP
zmienić taśmę
CSAVE TEMP,"ZBIÓR"
ERA TEMP
AMSDOS
uwaga 1</td><td></td></tr>
</table>

KOPIOWANIE DO	AMSTRAD BASIC zbiór typu . BAS	KOPIOWANIE Z TAŚMY Zbiór ASCII typ nieokreślony	Zbiór binarny typu . BIN
zbiór AMSTRAD BASIC typu .BAS na dysku	|TAPE LOAD "ZBIÓR" |DISC SAVE "ZBIÓR"		
zbiór ASCII na dysku	|TAPE LOAD "ZBIÓR" |DISC SAVE "ZBIÓR",A	<włożyć dysk z CP/M 2.2> |CPM CLOAD "ZBIÓR" AMSDOS	
zbiór binarny typu .BIN na dysku			H=HIMEN |TAPE MEMORY <s>-1 LOAD "ZBIÓR" |DISC SAVE "ZBIÓR",B, <s>, <l>[, <r>] MEMORY H (uwaga 2)

uwaga 1: wymagane wolne miejsce na dysku na zbiór przejściowy "TEMP"

uwaga 2: <s> oznacza adres początkowy zbioru, <l> oznacza długość zbioru, <r> to opcjonalny adres startowy

KOPIOWANIE Z DYSKU

<table>
<tr><th>KOPIOWANIE DO</th><th>AMSTRAD BASIC
zbiór typu .BAS</th><th>zbiór ASCII
typ nieokreślony</th><th>zbiór binarny
typu .BIN</th><th>inne zbiory
dyskowe</th></tr>
<tr><td>zbiór AMSTRAD
BASIC typu
.BAS na taśmie</td><td>LOAD "ZBIÓR"
|TAPE
SAVE "ZBIÓR"
|DISC</td><td></td><td></td><td></td></tr>
<tr><td>zbiór binarny
typu .BIN
na taśmie</td><td></td><td></td><td>H=HIMEM
MEMORY ⟨s⟩-1
LOAD "ZBIÓR"
|TAPE
SAVE "ZBIÓR"
|TAPE
SAVE "ZBIÓR",B,
⟨s⟩,⟨l⟩[, ⟨r⟩]
|DISC
MEMORY H
(uwaga 2)</td><td></td></tr>
<tr><td>zbiór ASCII
na taśmie</td><td>LOAD "ZBIÓR
|TAPE
SAVE "ZBIÓR",A
|DISC
lub
⟨włożyć dysk z CP/M 2.2⟩
|CPM
CSAVE ZBIÓR
AMSDOS</td><td>⟨włożyć dysk z CP/M 2.2⟩
|CPM
CSAVE ZBIÓR
AMSDOS</td><td></td><td>(włożyć dysk z CP/M 2.2)
|CPM
CSAVE ZBIÓR
AMSDOS
(uwaga 3)</td></tr>
</table>

KOPIOWANIE DO	KOPIOWANIE Z DYSKU AMSTRAD BASIC zbiór typu . BAS	zbiór ASCII typ nieokreślony	zbiór binarny typu . BIN	inne zbiory dyskowe
zbiór AMSTRAD BASIC typu .BAS na dysku	LOAD "ZBIÓR" (zmienić dysk) SAVF "ZBIÓR" lub (włożyć dysk z CP/M Plus) |CPM PIP B:=ZBIÓR AMSDOS			
zbiór ASCII na dysku	LOAD "ZBIÓR" (zmień dysk) SAVE "ZBIÓR",A	(włożyć dysk z CP/M Plus) |CPM PIP B:=ZBIÓR AMSDOS		
zbiór binarny typu .BIN na dysku			(włożyć dysk z CP/M Plus) |CPM PIP B:=ZBIÓR AMSDOS	
inny zbiór dyskowy				(włożyć dysk z CP/M Plus) |CPM PIP B:=ZBIÓR AMSDOS

uwaga 3: otrzymany zbiór nie może być bezpośrednio używany przez BASIC; sposób stosowany w celu przenoszenia zbiorów lub ich przechowywania. Zbiór może być skopiowany z powrotem na dysk przez CLOAD "ZBIÓR" w systemie CP/M 2.2.

Błędy sygnalizowane przez system AMSDOS

Jeżeli wprowadzony rozkaz nie może być przez system AMSDOS wykonany, wyświetlany jest komunikat o błędzie. Jeżeli występuje błąd w działaniu sprzętu, komunikat o błędzie zakończony jest pytaniem:

Retry, Ignore or Cancel ?
(Powtórz, Pomiń lub Odwołaj)

Na pytanie to odpowiedzieć można znakami R, I lub C.
R powoduje powtórzenie operacji, najlepiej po wykonaniu przez użytkownika zabiegów, usuwających przyczynę wystąpienia błędnego działania (np. włożenie dysku do napędu, o czym wcześniej użytkownik zapomniał).
I powoduje kontynuowanie wykonywanego programu (zadania) z pominięciem przerwanej przez błąd operacji; prowadzi to jednak najczęściej do nieoczekiwanego i nienormalnego rezultatu.
C powoduje odwołanie prowadzonej operacji i powrót do stanu przed jej zainicjowaniem.

Znaczenie komunikatów o błędach

Unknown command
..."Nierozpoznane polecenie", popełniono błąd przy wprowadzaniu rozkazu

Bad command
..."Polecenie nieprawidłowe", pomimo rozpoznania rozkazu przez system nie może on być z jakiś przyczyn wykonany. Błąd składni, niewłaściwy parametr lub niewłaściwa konfiguracja sprzętowa

<nazwa zbioru> already exists
..."Zbiór już istnieje", użytkownik próbuje zmienić nazwę zbioru na nazwę zbioru już istniejącego

<nazwa zbioru> not found
..."zbiór nieznaleziony", wywoływany zbiór nie istnieje

Drive <drive> : directory full
..."Skorowidz dysku <drive> całkowicie zapełniony", nie ma możliwości wprowadzenia do skorowidza nazwy nowego zbioru

Drive <drive> : disc full

..."Dysk <drive> całkowicie zapełniony", nie ma miejsca na nowy zbiór

Drive <drive> : disc changed, closing <nazwa zbioru>

..."Dysk <drive> został zmieniony, zamykając zbiór <nazwa zbioru> ", zmieniono dysk z ciągle otwartymi zbiorami na inny

<nazwa zbioru> is read only

... "zbiór jest tylko do czytania", nie ma możliwości zapisu do zbioru. Zbiór może być określony jako Read/Only lub jako dostępny do odczytu i zapisu (Read/Write) tylko za pomocą systemu CP/M.

Drive <drive> : disc missing

..."Dysk <drive> nie znaleziony", nie ma dysku w napędzie dyskowym lub jest włożony nieprawidłowo. Należy prawidłowo wprowadzić dysk, a następnie nacisnąć klawisz R.

Drive <drive> : disc is write protected

..."Dysk <drive> : zabezpieczony przed zapisem", próba zapisu dysku z otwartym otworem zabezpieczającym. Wyjąć dysk, zasłonić otwór zabezpieczający, włożyć dysk do napędu i nacisnąć klawisz R.

Drive <drive> : read fail

..."Dysk <drive> : błąd odczytu", błąd sprzętu przy odczycie dysku. Zaleca się wyjąć i powtórnie wprowadzić dysk do napędu, a następnie nacisnąć klawisz R.

Drive <drive> : write fail

..."Dysk <drive> : błąd zapisu", błąd sprzętu przy zapisie dysku. Zaleca się wyjąć i powtórnie wsunąć dysk do napędu, a następnie nacisnąć klawisz R.

Failed to load CP/M

..."Nieudane ładowanie systemu CP/M", błąd odczytu podczas wprowadzania systemu po rozkazie |CPM lub użycie niewłaściwego dysku systemowego. Próba ładowania systemu CP/M z dysku, sformatowanego jako niesystemowy (Data format) powoduje zgłoszenie błędu odczytu (read fail error).

Część 2: CP/M

Omawiane zagadnienia:
- Wprowadzenie do CP/M
- Uruchomienie CP/M Plus
- Polecenia rezydentne
- Polecenia nierezydentne
- Zarządzanie urządzeniami peryferyjnymi
- Praca z CP/M 2.2

CP/M Plus to dyskowy system operacyjny. Wysoka efektywność tego systemu daje Ci dostęp do całej potęgi Twojego CPC 6128, w pełni wykorzystując 128K-bajtową pamięć RAM z ponad 61 KB przeznaczonymi na programy użytkowe . CP/M zapewnia swobodny dostęp do zbiorów danych, a implementacja dostosowana do CPC 6128 zawiera skuteczny emulator zespołu monitora ekranowego.

CP/M jest szeroko stosowany w bardzo wielu komputerach i istnieją tysiące różnych programów użytkowych, które mogą być dostępne dla Ciebie wraz z całym związanym z nim bogactwem wiedzy i doświadczenia.

Szczegółowy opis systemu CP/M Plus wraz z informacjami jak pisać własne programy użytkowe oraz omówieniem implementacji systemu w komputerze AMSTRAD zawiera opracowanie SOFT971-A Guide to CP/M Plus (Przewodnik po CP/M Plus) a także inne publikacje firmy AMSOFT lub innych wydawców.

Wstęp

System operacyjny CP/M umożliwia Ci komunikowanie się z komputerem, manipulowanie zbiorami i sterowanie urządzeniami peryferyjnymi. Rozkazy systemowe i umieszczone na dyskach programy użytkowe pomogą Ci wykonać zadanie, jakim jest napisanie i uruchomienie własnych programów, dostosowanych do Twoich potrzeb i operujących Twoimi danymi. Możesz nawet stać się ekspertem w zakresie działania CP/M i różnych programów użytkowych i czasami Twoje doświadczenie może być pomocne dla nas w rozwiązywaniu wspólnych problemów. Na początek trzeba wiedzieć tylko jak zacząć i reszta tego rozdziału zawiera wstępne

informacje o możliwościach i zaletach systemu bez zaciemniania istotnych informacji zbyt wielką liczbą szczegółów.

Podobnie jak BASIC, który pracować może w trybie bezpośrednim, sygnalizowanym napisem "Ready", CP/M pracuje w trybie bezpośredniego wykonywania poleceń operatorskich, jeżeli wyświetlany jest znak zachęty A> lub B> . W tym trybie system wykonuje jedynie kilka podstawowych poleceń rezydentnych. Ogromna większość poleceń systemowych wykonywana jest jako tzw. "polecenia przejściowe" - nierezydentne. Nazwa ta wynika z tego, że programy realizujące takie polecenie zostają wprowadzone do pamięci komputera z dysku tylko w razie konieczności ich wykonywania. Programy realizujące polecenia rezydentne są natomiast wbudowane na stałe w zasadniczy systemowy program operacyjny.

System sygnalizuje błędy w formie standardowych komunikatów systemu CP/M a ponadto sygnalizuje szereg błędów związanych ze sprzętem w wyróżnionej przesuwającej się najniższej linii ekranu.

CP/M Plus na dysku

Zasadnicza część systemu CP/M Plus znajduje się w specjalnym zbiorze typu ".EMS" na Stronie 1 kompletu dysków systemowych System ładowany jest z tego zbioru do pamięci komputera w dwustopniowym procesie wprowadzania.

Po komendzie |CPM w systemie AMSDOS wprowadzana jest najpierw do pamięci zawartość pierwszego sektora ścieżki 0. Na dyskach systemowych sektor ten zawiera program ładujący do pamięci komputera zbiór typu .EMS. Pozostałe sektory ścieżek systemowych są puste (nieużywane).

Profilowanie parametrów przy starcie

Po zakończeniu procesu ładowania, podczas inicjalizacji pracy systemu CP/M Plus poszukiwany jest na dysku zbiór PROFILE.SUB. Jeżeli zbiór taki istnieje, wykonywane są zapisane w nim instrukcje. Można to wykorzystać do przeprogramowania klawiatury, zmiany sposobu wyświetlania na ekranie, inicjaliza-

Kod kontrolny	Klawisz	Działanie
[CONTROL] A		Przesuwa kursor o jeden znak w lewo
[CONTROL] B	[CONTROL] ← lub [CONTROL] →	Przesuwa kursor do początku linii. Jeżeli kursor jest już na początku linii, przesuwa kursor do końca linii
[CONTROL] C	[CONTROL] [ESC]	Zaniechanie wykonywanej operacji
[CONTROL] E	[CONTROL] [RETURN]	Przenosi dalszy ciąg linii do początku następnej linii /fizyczny koniec linii/
[CONTROL] F	→	Przesuwa kursor o jeden znak w prawo
[CONTROL] G	[CLR]	Kasuje znak wskazywany przez kursor
[CONTROL] H	[DEL]	Kasuje znak w miejscu kursora i cofa kursor o jeden znak w lewo (Backspace)
[CONTROL] I	[TAB]	Przesuwa kursor do następnej pozycji tabulacji
[CONTROL] J		Kończy wprowadzanie linii rozkazu
[CONTROL] K	[CONTROL] [CLR]	Kasuje do końca linii
[CONTROL] M	[RETURN] lub [ENTER]	Kończy wprowadzanie linii rozkazu
[CONTROL] P		Włącza lub wyłącza wyprowadzanie znaków jednocześnie na ekran i drukarkę
[CONTROL] Q		Wznawia zatrzymane wcześniej wyprowadzanie znaków na ekran
[CONTROL] R	[CONTROL] [ENTER]	Wypisuje ponownie wprowadzoną linię rozkazu
[CONTROL] S	[ESC]	Wstrzymuje wyprowadzanie znaków na ekran; wznowienie wyprowadzania za pomocą [CONTROL] Q
[CONTROL] U		Anuluje wprowadzoną linię
[CONTROL] W	[COPY]	Wywołuje ostatnio wprowadzoną linię rozkazową
[CONTROL] X	[CONTROL] [DEL]	Kasuje linię od początku do pozycji kursora
[CONTROL] Z		Koniec tekstu

Nazwy zbiorów

Wiele rozkazów używa jako parametr nazwy zbiorów. W przypadku określonych rozkazów nazwy zbiorów zawierać mogą znaki zastępcze (patrz punkt "Znaki zastępcze" w części 1 tego rozdziału). Wszystkie nazwy zbiorów przyjmowane są tak, jak by były pisane dużymi literami.

W poleceniach rezydentnych i większości programów użytkowych nie jest wymagane pisanie nazwy zbiorów w cudzysłowach " ". Nazwa zbioru może być poprzedzona symbolem A: lub B:, określającym użycie odpowiedniego napędu dyskowego.

Typowym rozkazem CP/M jest np.:

TYPE KEYS.CCP

gdzie TYPE oznacza polecenie wyświetlenia na ekranie a KEYS.CCP określa nazwę zbioru, którego treść ma być wyświetlana.

Przełączanie dysków

W przypadku dołączenia drugiego napędu dyskowego, można ustanowić domyślne przyjmowanie pracy z napędem A albo B przez wprowadzenie polecenia A: lub B: po znaku zachęty /gotowości/ B> albo A>. Wyświetlany znak gotowości pokazuje, który dysk jest aktualnie domyślnie wybierany. Dodanie przedrostka A: lub B: przed nazwą zbioru może nakazać pracę z innym napędem, nie zmieniając jednakże dysku przyjmowanego domyślnie w przypadku gdy nie podano symbolu dysku przed nazwą zbioru.

Polecenia rezydentne /bezpośrednie/

System CP/M Plus wykonywać może szereg poleceń bezpośrednich /rezydentnych/, wprowadzanych po znaku zachęty A> lub B>. Nazwę polecenia podawać można w formie skrótowej. Opisane poniżej proste funkcje wbudowane są w stale rezydujący w pamięci komputera system operacyjny, tą samą nazwą wywołać można jednak także bardziej złożoną funkcję nierezydentną, realizowaną przez dodatkowe programy wprowadzane z dysku systemowego.

DIR

DIR powoduje wyświetlenie skorowidzu (DIRectory) dysku. Nazwy zbiorów nie są przy tym porządkowane w żaden szczególny sposób, a kolejność ich wyprowadzania odpowiada kolejności umieszczenia w skorowidzu. W nazwach zbiorów dołączanych jako parametr dyrektywy stosować można znaki zastępcze. DIR nie wyświetla nazw zbiorów z nadanym atrybutem SYS

DIR	wyświetla skorowidz przyjmowanego domyślnie dysku
DIR B:	wyświetla skorowidz dysku B:
DIR *.BAS	podaje listę zbiorów typu .BAS
DIR B:*.BAS	podaje listę zbiorów typu .BAS z dysku B:
DIR PIP.COM	wprowadza tylko nazwę zbioru PIP.COM (o ile zbiór taki występuje w skorowidzu)

DIRSYS lub DIRS

DIRSYS lub DIRS wyświetla listę występujących w skorowidzu zbiorów z nadanym atrybutem SYS . Poza tym działa identycznie jak DIR. Atrybut SYS opisano nieco dalej.

ERASE lub ERA

Usuwa (ERAse) nazwę zbioru ze skorowidzu dysku. Sam zbiór pozostaje nadal na dysku, dopóki w jego miejsce nie zostanie wpisany nowy zbiór, jednakże w wyniku usunięcia go ze skorowidzu staje się już niewidoczny /niedostępny/. W podawanych jako parametr dyrektywy nazwach zbiorów stosować można znaki zastępcze, jednakże w takim przypadku wyprowadzane jest żądanie potwierdzenia dyrektywy. ERA nie listuje nazw kasowanych zbiorów. Jeżeli jakikolwiek zbiór, który ma być usunięty okazuje się zbiorem zabezpieczonym przed zapisem /"tylko do czytania" - Read/Only/, to przerywane jest wykonywanie dyrektywy. Atrybut "tylko do czytania" jest opisany nieco dalej.

ERA PIP.COM	usuwa zbiór PIP.COM
ERA B:PIP.COM	usuwa zbiór PIP.COM z dysku B:
ERA *.BAS	usuwa wszystkie zbiory typu .BAS

RENAME lub REN

REN zmienia nazwę zbioru (REName). Podaje się najpierw nową nazwę zbioru a następnie znak = i starą nazwę zbioru. Jeżeli nowa nazwa już istnieje, sygnalizowany jest błąd.

Polecenie rezydentne REN nie dopuszcza stosowania znaków zastępczych w nazwach zbiorów; użycie takich znaków powoduje wywołanie nierezydentnego programu RENAME.COM

REN NEWNAME.BAS = OLDNAME.BAS	zmienia nazwę zbioru z OLDNAME.BAS na NEWNAME.BAS
REN B:NEWNAME.BAS = OLDNAME.BAS	zmienia nazwy zbiorów na dysku B:

TYPE lub TYP

Wyświetla na ekranie treść /zawartość/ zbioru. Zbiór musi być zbiorem tekstowym ASCII; w innym przypadku wystąpić mogą nieprzewidziane i niepożądane efekty.

TYPE KEYS.CCP

wyświetla zawartość zbioru KEYS.CCP

USER lub USE

USER zmienia numer aktualnego użytkownika. Przy inicjalizacji systemu CP/M Plus system przyporządkowany zostaje użytkownikowi nr. 0. System zapewnia normalny dostęp tylko do zbiorów aktualnie określonego użytkownika; umożliwia to podział dysku /i skorowidzu dyskowego/ między kilku różnych użytkowników.

Zbiory Użytkownika Nr. 0 z przyporządkowanym atrybutem 'SYS' są dostępne dla wszystkich użytkowników. Pozwala to na korzystanie z programów systemowych i aplikacyjnych przez różnych użytkowników, bez konieczności kopiowania tych programów do obszarów dyskowych poszczególnych użytkowników.

USER 3

przyporządkowywuje system użytkownikowi nr 3.

Polecenia nierezydentne

Do prowadzenia bardziej złożonych operacji na zbiorach, niż umożliwiają to polecenia rezydentne, należy użyć jednego z licznych dostarczonych programów użytkowych. Programy takie wywoływane są przez proste napisanie ich nazwy, za którą w razie potrzeby umieszcza się nazwę zbioru i/lub dodatkowe parametry. Prawdopodobnie wywoływałeś już w ten sposób program DISCKIT 3.

Polecenia nierezydentne należą do jednej z licznych, opisanych niżej grup. Pełna dokumentacja tych programów jest bardzo obszerna i dalsze informacje znaleźć można w zbiorze HELP (na 3 Stronie pakietu dysków systemowych) oraz w opracowaniu SOFT971-A Guide to CP/M Plus.

Programy DISCKIT 3, SETKEYS, SETLST, SETSIO, PALETTE, LANGUAGE i AMSDOS, pakiet sterowania ekranem GSX oraz zaimplementowana wersja LOGO 3 zostały opracowane przez firmę AMSTRAD i są przeznaczone do stosowania wyłącznie w komputerach tej firmy. Programy te nie będą działały w żadnym innym komputerze z systemem CP/M.

Jest możliwe wprowadzenie kilku poleceń w jednej linii rozkazowej, rozdzielając poszczególne polecenia znakiem wykrzyknika, np.:

LANGUAGE 3 ! SETKEYS KEYS.WP

Zarządzanie urządzeniami peryferyjnymi

DISCKIT 3 jest programem, umożliwiającym formatowanie, kopiowanie i sprawdzanie dysków. Kopiowanie zbiorów na dysk czysty (niesformatowany) poprzedzone jest formatowaniem, co jest szybsze niż osobne formatowanie i kopiowanie. Wyświetlany na ekranie zbiór poleceń (menu) podaje, którymi klawiszami (głównie spośród wydzielonych klawiszy funkcyjnych) steruje się działaniem programu. Program umożliwia stosowanie specjalnego formatu sprzedażnego (Vendor format), używanego przy dystrybucji oprogramowania, chociaż formatowanie systemowe (Data format) jest być może bardziej odpowiednie do tego celu w systemie CP/M Plus.

OSTRZEŻENIE

Obowiązująca nabywcę umowa licencyjna na system CP/M Plus (zawierający zakodowany elektronicznie numer seryjny) zezwala na używanie tego systemu tylko w jednym, własnym komputerze. Oznacza to, iż w szczególności zabronione jest przekazywanie innym osobom dysków z kopią systemu CP/M, oznaczoną TWOIM numerem seryjnym. Ponieważ kopie Strony 1 pakietu dysków systemowych zawierają w zbiorze typu .EMS Twój CP/M, uważaj, aby nie sprzedać, wymienić lub w jakikolwiek inny sposób rozstać się z dyskiem, zawierającym taki zbiór.

Języki alfabetu

CPC 6128 zawiera zestaw znaków alfabetu kilku różnych języków. Polecenie LANGUAGE (język) zmienia niektóre znaki, wyświetlane na ekranie, umożliwiając używanie znaków stosowanych w różnych alfabetach. Więcej szczegółów na ten temat zamieszczono w części 16 rozdziału "Nieco użytecznych informacji"

Polecenie:

LANGUAGE 3

wprowadza np. znaki języka angielskiego, stosowane w Wielkiej Brytanii, co powoduje zamianę znaku funta na £ zamiast #, jaki występuje w używanym normalnie w komputerze zestawie znaków.

Kolory

Po inicjalizacji systemu CP/M Plus w komputerze 6128 z monitorem kolorowym wyświetlane są na ekranie intensywnie białe litery na niebieskim tle. Kolory te można zmienić za pomocą dyrektywy PALETTE z parametrami, określającymi barwy "atramentów": pierwszy dotyczy barwy tła (atrament nr 0) a drugi liter (atrament nr 1). Kolory określone są liczbami z zakresu od 0 do 63; te same parametry mogą być użyte do zmiany jaskrawości wyświetlania tekstu na ekranie monitora monochromatycznego (zielonego).

Można także określić kolory większej liczby atramentów aż do szesnastu, jednakże tylko dwa pierwsze są widoczne w ustawionym trybie wyświetlania 80-ciu znaków w wierszu

Polecenie:

PALETTE 63,1

zmienia kolory wyświetlania na przeciwne, tzn. niebieskie (1) litery na intensywnie białym (63) tle.

W tabeli podano liczby określające poszczególne kolory (lub ich intensywności). Stosować można zarówno wartości dziesiętne jak i szesnastkowe, w zależności od upodobania.

Kolor	Szesnastkowo	Dziesiętnie	Kolor	Szesnastkowo	Dziesiętnie
Czarny	&00	0	Pastelowoniebieski	&2B	43
Niebieski	&02	2	Pomarańczowy	&2C	44
Jaskrawoniebieski	&03	3	Różowy	&2E	46
Czerwony	&08	8	Pastelowoszkarłatny	&2F	47
Szkarłatny	&0A	10	Jaskrawozielony	&30	48
Bladofioletowy	&0B	11	Zieleń morska	&32	50
Jaskrawoczerwony	&0C	12	Jaskrawolazurowy	&33	51
Purpurowy	&0F	14	Cytrynowozielony	&38	56
Jaskrawoszkarłatny	&0F	15	Pastelowozielony	&3A	58
Zielony	&20	32	Jasnolazurowy	&3B	59
Lazurowy	&22	34	Jaskrawożółty	&3C	60
Błękitny	&23	35	Pastelowożółty	&3E	62
Żółty	&28	40	Jaskrawobiały	&3F	63
Biały	&2A	42			

Klawiatura

Kody wysyłane przez klawiaturę mogą być zmieniane za pomocą polecenia SETKEYS. Umożliwia to przypisywanie różnych kodów do poszczególnych klawiszy. Żądane kody muszą być wpisane w zbiór, którego nazwę podaje się jako parametr dyrektywy SETKEYS. Zbiór taki można utworzyć za pomocą edytora tekstu, programu PIP, lub także w języku BASIC. Dla przykładu:

SETKEYS KEYS.TST

gdzie zbiór KEYS.TST zawiera:

E &8C "DIR"↑M" expansion token 12

N S C "↑H" backspace = [CONTROL]H, ASCII08

przypisuje klawiszom CONTROL ENTER (reprezentowanym przez 8C) wprowadzenie polecenia DIR RETURN oraz klawiszowi przesuwającemu kursor w lewo (klawisz nr. 8) kod funkcji "backspace" - kasowanie znaku w miejscu kursora i przesunięcie kursora w lewo.

W pakiecie dysków systemowych 6128 znajduje się zbiór KEYS.CCP z kodami klawiatury wymaganymi w systemie CP/M, zbiór KEYS.DRL do użycia przy pracy z Dr. LOGO (ze Strony 3) oraz KEYS.WP, odpowiedni dla wielu procesorów tekstu (word processors).

Drukarka i ekran

Inicjalizację drukarki można przeprowadzić za pomocą polecenia

SETLST <nazwa zbioru>

gdzie zbiór nazwa zbioru zawiera ciąg znaków, wysyłany do drukarki. Podobnie jak w zbiorze poleceń dla dyrektywy SETKEYS, znaki kontrolne mogą być reprezentowane przez:

↑<znak>

lub ↑<wartość znaku>

lub ↑<nazwa funkcji>

gdzie nazwy funkcji to ESC, FF i inne według tabeli kodów ASCII, zamieszczonej w rozdziale "Trochę użytecznych informacji".

Często używanym kodem inicjalizacyjnym dla wielu różnych drukarek jest kod o wartości 15 nakazujący drukowanie małą czcionką. Kod taki można wysłać w języku BASIC za pomocą rozkazu

PRINT #8, CHR$(15)

W systemie CP/M to samo polecenie ma postać

SETLST CONDENSE

gdzie zbiór CONDENSE zawierać musi jedną z poniżej podanych pojedynczych linii tekstu:

↑'SI'
↑O
↑'&F'
↑'15'

Każda z tych linii określa wartość dziesiętną 15.

Niektóre programy aplikacyjne wymagają użycia ekranu o 24 wierszach po 80 znaków. Takie wymiary ekranu można ustawić za pomocą polecenia SET24X80.

Rozkaz:

SET24X80

lub

SET24X80 ON

włącza tryb wyświetlania 24 x 80 znaków, a:

SET24X80 OFF

wyłącza ten tryb pracy.

Normalnie, w 6128 używany jest ekran o wymiarach 24 x 80 z najniższą linią zarezerwowaną na komunikaty systemowe. Wyłączenie trybu 24 x 80 można zauważyć tylko w przypadku jednoczesnego wyłączenia pracy linii systemowej. Sposób wyłączenia tej linii opisano w części 15 Rozdziału 7.

Kanał transmisji szeregowej

System CP/M Plus zawiera procedury umożliwiające korzystanie z pojedynczego szeregowego kanału wejścia/wyjścia. Parametry transmisji przez ten kanał można odczytać za pomocą polecenia SETSIO (bez żadnego dodanego parametru):

SETSIO

lub ustawić za pomocą polecenia, zawierającego wszystkie lub wybrane parametry z:

SETSIO, RX 1200, TX 75, PARITY NONE, STOP 1, BITS 3,
HENDSHAKE ON, XOFF OFF

Szybkość transmisji (liczbę bodów) oraz używanie lub nieużywanie specjalnych znaków rozpoznawczych (status XON/XOFF) można także deklarować za pomocą odpowiedniej dyrektywy DEVICE

(urządzenie). DEVICE przypisuje urządzeniom logicznym odpowiednie urządzenia fizyczne. Urządzenia logiczne są oznaczane znakiem dwukropka : . Aktualne atrybuty urządzeń można poznać, wprowadzając polecenie:

DEVICE

a zmianę atrybutów uzyskać za pomocą poleceń takich jak:

DEVICE SIO [1200] - ustawia szybkość transmisji szeregowej 1200 bodów

DEVICE SIO [XON] - włącza protokół XON/XOFF w kanale transmisji szeregowej

DEVICE SIO [NOXON] - wyłącza protokół XON/XOFF

Można także zmieniać przyporządkowanie urządzeń logicznych fizycznymi. Normalnie urządzeniu logicznemu CON: (CONsole - konsola operatorska) jest przypisane urządzenie fizyczne CRT (Cathode Ray Tube - monitor telewizyjny, klawiatura), urządzeniu logicznemu AUX: (AUXillary - pomocnicze) kanał fizyczny transmisji szeregowej SIO (Serial Input Output - szeregowe wejście-wyjście) a urządzeniu logicznemu LST: (LiST - urządzenie listujące) urządzenie fizyczne LPT (Line PrinTer - drukarka liniowa, dołączana przez sprzęg równoległy typu "Centronics"). Polecenie:

DEVICE LST:=SIO

powoduje wysyłanie znaków przeznaczonych dla drukarki do wyjścia szeregowego.

Proszę zauważyć występującą przy tym zmianę wykorzystania strumieni przesyłania informacji, ułatwi to zrozumienie możliwości kopiowania zbiorów, oferowanych przez program PIP. Podobnie, zmianę strumieni przesyłowych realizuje się za pomocą dyrektyw GET ⟨nazwa zbioru⟩ i PUT ⟨nazwa zbioru⟩ , które powodują pobieranie informacji ze zbioru, zamiast z konsoli operatorskiej i odsyłanie informacji do zbioru, zamiast do konsoli (monitora) lub drukarki (GET oznacza "dostawać" a PUT "umieszczać")

PIP

PIP (Peripheral Interchange Program) to program użytkowy, umożliwiający przesyłanie informacji między różnymi urządzeniami peryferyjnymi komputera.
Zasadnicza składnia polecenia jest następująca:

PIP <odbiorca> = <źródło>

Jako <odbiorca> i <źródło> może być zadeklarowana nazwa zbioru z dopuszczalnymi w nazwie źródła znakami zastępczymi albo jedno z podanych poniżej urządzeń logicznych:

jako źródło:		jako odbiorca:	
CON:	konsola operatorska/ klawiatura	CON:	konsola operatorska/ monitor
AUX:	wejście pomocnicze	AUX:	wyjście pomocnicze
EOF:	/End-Of-File/, znak końca zbioru	PRN:	drukarka, przy czym przy transmisji rozwijane są odpowiednio znaki tabulacji, dodawana numeracja wierszy wydruku i wysyłane znaki przesuwania stron druku

Przykładowe dyrektywy PIP to:

PIP B:=A:*.COM

- kopiuje wszystkie (*) zbiory typu .COM z dysku A: na dysku B:

PIP KEYBOARD.CPM=KEYS.CCP

- sporządza kopię zbioru KEYS.CCP, nadając mu nazwę KEYBOARD.CPM

PIP CON:=KEYS.CCP

- wysyła zbiór KEYS.CCP na ekran (podobnie jak polecenie TYPE KEYS.CCP)

PIP TYPEIN.TXT=CON:

- umieszcza tekst wprowadzany za pomocą klawiatury w zbiorze o nazwie TYPEIN.TXT

Ostatnią operację należy zakończyć przez wprowadzenie kodu [CONTROL] Z (Znak EOF - koniec zbioru) a w celu przesunięcia się do początku nowej linii należy dodać znak [CONTROL] I po każdym kończącym linię znaku [RETURN]. Znak [CONTROL] I oznacza w ASCII polecenie przesunięcia do następnej linii (Line Feed).

Wprowadzenie PIP bez żadnych parametrów powoduje wyświetlenie znaku *, po którym można wprowadzać dalsze żądane polecenia.

Jest to szczególnie korzystne przy kopiowaniu zbiorów z dysku na dysk w przypadku, gdy żaden z dysków nie zawiera zbioru PIP.COM. Można wówczas wprowadzić program PIP ze strony 1 dysku systemowego, usunąć dysk systemowy i wprowadzić dyski wymagane przy kopiowaniu.

Aby zakończyć pracę programu PIP, należy nacisnąć [RETURN] bezpośrednio po znaku * .

PIP może być używany do kopiowania zbiorów z jednego dysku na drugi także w systemie z jednym napędem dyskowym, zgłaszając w odpowiednim momencie żądanie wymiany dysku. Dysk źródłowy i dysk docelowy muszą być przy tym oznaczone innymi identyfikatorami.

Administrowanie zbiorami

DIR, ERASE, RENAME i TYPE są programami systemowymi, o znacznie większych możliwościach, niż takie same funkcje rezydentne (wbudowane). Jak w wielu innych programach systemowych, opracowanych przez Digital Research, dodatkowe parametry tych funkcji podaje się w nawiasach kwadratowych. Ich pełny opis zawiera zbiór HELP (na Stronie 3 pakietu dysków systemowych). Przykładowe polecenia to:

DIR [FULL]	podaje rozmiary i atrybuty zbiorów
ERASE *.COM [CONFIRM]	żąda potwierdzenia przy każdym znalezionym zbiorze
RENAME	żąda kolejno wprowadzenia starej i nowej nazwy zbioru
RENAME *.SAV= *.BAK	zmienia wszystkie zbiory typu .BAK na zbiory typu .SAV
TYPE KEYS.WP [NOPAGE]	pomija stronicowanie wyświetlanej na ekranie treści zbioru

O atrybutach zbiorów SYS (System - zbiór systemowy) i RO (Read/Only - zbiór "tylko do czytania") wspomniano już wcześniej. Te i inne atrybuty mogą być nadawane za pomocą dyrektywy SET (ustaw), przy czym mogą być także stosowane znaki zastępcze w nazwach zbiorów.

Polecenia:

SET *.COM [RO]

SET KEYS.CCP [RO]

SET A: [RO]

nadają zbiorom na dysku status "zbiorów tylko do czytania" (Read/Only), co zabezpiecza te zbiory przed przypadkowym skasowaniem.

Polecenia:

SET *.COM [RW]

SET KEYS.CCP [RW]

SET A: [RW]

przywracają zbiorom na dysku status zbiorów do odczytu i zapisu (Read/Write)

Polecenia:

SET *.COM [SYS]

SET KEYS.CCP [SYS]

przypisują zbiorom atrybut zbiorów systemowych. Zbiory z takim atrybutem nie są wykazywane przez dyrektywę DIR (lecz tylko przez dyrektywę DIRS lub DIRSYS). Jednakże zbiory takie są dostępne do użytku, a co więcej, zbiory SYS przyporządkowane użytkownikowi nr. 0 mogą być wykorzystywane przez wszystkich innych użytkowników.

Polecenia

SET *.SOM [DIR]

SET KEYS.CCP [DIR]

odbierają zbiorom atrybut zbiorów systemowych.

Każdemu dyskowi można przypisać nazwę (etykietę) (NAME) i hasło (PASSWORD). Hasło chroni skorowidz dysku, nie chroni natomiast zbiorów, wymienionych w skorowidzu. Poszczególnym zbiorom można także przydzielić hasło.

Polecenia

SET [NAME = ROLAND]

SET [PASSWORD=SALLY]

SET [PROTECT=ON]

oddziaływują na przymowany domyślnie dysk.

Polecenia:

SET *.* [PASSWORD=SALLY]

SET *.* [PROTECT=READ]

zabezpieczają zbiory na przyjmowanym domyślnie dysku (użyte tutaj znaki zastępcze *.* oznaczają "wszystkie zbiory").

Polecenie INITDIR (na Stronie 2 pakietu dysków systemowych) aktywuje procedurę wpisywania do skorowidzu dysku daty zakładania (CREATE), modyfikowania (UPDATE) i wywoływania (ACCES) zbiorów.

Polecenia:

INITDIR
SET [CREATE=ON] lub SET [ACCESS=ON] i
SET [UPDATE=ON] oraz
DIR [FULL]

inicjalizują wpisywanie i wyświetlają daty dla przyjmowanego domyślnie dysku. Aktywizacje wpisywania dat wymaga ustawienia daty za pomocą dyrektywy:

DATE SET

przy każdorazowej inicjalizacji systemu CP/M Plus. Po inicjalizacji data i czas jest odmierzana samoczynnie przez umieszczony w 6128 zegar. Stan zegara można odczytać za pomocą dyrektywy:

DATE
lub DATE CONTINUOUS

OSTRZEŻENIE:
Jeżeli na dysk wpisano nazwę dysku, hasło lub daty to na dysk taki NIE WOLNO dopisywać żadnych zbiorów w systemie AMSDOS lub CP/M 2.2, gdyż żaden z tych systemów nie potrafi należycie obsłużyć takich dysków.

W normalnym trybie pracy system wyszukuje zadane zbiory tylko na jednym, określonym w nazwie lub przyjmowanym domyślnie dysku. Polecenie:

SETDEF *,A

(gdzie * oznacza dysk domyślny) powoduje wyszukiwanie zbiorów najpierw na przyjmowanym domyślnie dysku a następnie na dysku A: . Jeżeli przy tym domyślnym dyskiem jest dysk B:, to zadane zbiory zostaną znalezione nawet wtedy, gdy występują na dysku A:

Polecenia:

SETDEF [PAGE]

i SETDEF [NOPAGE]

włączają lub wyłączają automatyczne stronicowanie treści, wyświetlanej na ekranie (PAGE - stronica).

Należy pamiętać, że większość udogodnień, wprowadzanych przez rozkazy DEVICE, SET i SETDEF, szczególnie gdy dotyczą stałych urządzeń, a nie określonych zbiorów lub dysków może być ustalana łącznie z datą przy każdej inicjalizacji systemu CP/M Plus. Stanowić to może główne zastosowanie odpowiedniego zbioru PROFILE.SUB.

Program SUBMIT umożliwia automatyczne wykonywanie ciągu poleceń zapisanych w zbiorze typu .SUB. Polecenia w takim zbiorze zapisywane są zwykłym tekstem. Można w nim zamieścić polecenia dla wywoływanych programów, jeżeli pierwszy znak linii programowej jest znakiem typu <.

Pojemność dysku, pozostałe wolne miejsce, liczba wolnych pozycji skorowidzu, symbole użytkowników, posiadających zbiory na dysku i etykieta (nazwa) dysku (jeśli istnieje) mogą być wyświetlone za pomocą różnych wariantów dyrektywy SHOW (pokaż):

```
SHOW B:
SHOW B: [LABEL]
SHOW B: [USERS]
SHOW B: [DIR]
SHOW B: [DRIVE]
```

Wszystkie powyższe przykłady dotyczą dysku B:

Wyłączenie CP/M Plus

AMSDOS

Program ten wyłącza CP/M i przekazuje sterowanie wbudowanemu systemowi AMSTRAD BASIC; w systemie tym operacje dyskowe przeprowadza się za pomocą dyrektyw AMSDOS.

Wyższy poziom programowania

Strona 2 pakietu dysków systemowych zawiera liczne programy przeznaczone dla zaawansowanych programistów. Posługiwanie się nimi opisane jest w SOFR971 - A Guide to CP/M Plus i innych publikacjach.

Praca w systemie CP/M 2.2

W tym punkcie omówiono różnice, jakie występują przy używaniu systemu CP/M 2.2.

CP/M 2.2 jest ładowany z pierwszych dwóch ścieżek odpowiedniego dysku systemowego. Program ładujący system, umieszczony w pierwszym sektorze pierwszej ścieżki dysku jest inny, niż używany w przypadku CP/M Plus i należy uważać, aby zastosować właściwy dysk. Chociaż możliwe jest używanie dysków o formacie niesystemowym, jak format sprzedażny (Vendor format), format IBM i format danych (Data format), dyski o takim formacie można wykorzystywać jedynie w dodatkowym napędzie B:

Z wyjątkiem specjalnych programów (jak np. program FILECOPY), CP/M 2.2 umożliwia zapis na dysk tylko wtedy, gdy był on wcześniej "zaksięgowany" w systemie. Ponadto, format dysku (System, Data lub IBM) jest rozpoznawany tylko w czasie "księgowania". W przypadku głównego napędu dyskowego umieszczonego w komputerze (napędu dyskowego A:), "księgowanie" dysku ma miejsce w czasie powrotu do trybu bezpośredniego przyjmowania poleceń systemowych po zakończeniu pracy przejściowego programu użytkowego lub po naciśnięciu przy znaku zachęty A> lub B> klawiszy [CONTROL] C. W przypadku dodatkowego napędu dyskowego (napędu dyskowego B:) rozpoznawanie i "księgowanie" dysku ma miejsce przy pierwszym dostępie do dysku B: po "zaksięgowaniu" dysku A:.

Próba zapisu na dysk, który nie był wcześniej zaksięgowany, powoduje wyświetlenie informacji o błędzie w postaci:

Bdos Err on <dysk> : R/O

(Błąd dostępu do dysku, dysk tylko do czytania)

Praca systemu jest kontynuowana po naciśnięciu jakiegokolwiek klawisza. Wprowadzenie dysku o innym sposobie formatowania także powoduje sygnalizację błędu odczytu lub zapisu. Dla kontynuowania pracy systemu należy nacisnąć klawisz C.

W przypadku otrzymania oprogramowania na dysku o formacie sprzedażnym (Vendor format), w celu umożliwienia normalnego korzystania z zamieszczonych na nim programów należy przepisać te programy na dysk systemowy systemu CP/M 2.2 za pomocą FILE-COPY lub PIP lub ewentualnie wprowadzić na otrzymany dysk własny system CP/M 2.2, zamieniając go na dysk systemowy. Zrealizować to można za pomocą dyrektyw BOOTGEN i SYSGEN.

SYSGEN (bez żadnych dodatkowych parametrów) jest specjalnym programem kopiującym ścieżki z systemem CP/M 2.2 z jednego dysku na drugi; po wywołaniu program pyta się o dysk źródłowy i dysk odbierający kopiowany system. BOOTGEN kopiuje sektor 1 ścieżki 0 z programem ładującym i sektor, zawierający oprogramowanie zależne od konfiguracji sprzętowej komputera.

W rozkazie DIR nie są przyjmowane żadne parametry inne niż nazwy zbiorów. Wyprowadzane nazwy zbiorów nie są w jakikolwiek sposób uporządkowane i wyświetlane są zgodnie z kolejnością zajmowanych przez nie pozycji skorowidzu dyskowego.

STAT realizuje niektóre podstawowe funkcje poleceń SET i SHOW. Rozkazy:

```
STAT
STAT A:
STAT B:
```

wyprowadzają informacje o statusie i wolnym miejscu dysku

Polecenia:

```
STAT  *.COM
STAT  DISC.BAS
```

wyświetlają skorowidz żądanych zbiorów, uzupełniony dodatkowymi informacjami o tych zbiorach.

Polecenia:

```
STAT *.COM $R/O
STAT DISC.BAS $R/O
```

nadają zbiorom status Read/Only (tylko do czytania) tak, aby zbiory te nie mogły być przypadkowo skasowane lub zapisane zmienioną treścią.

Polecenia:

```
STAT *.COM. $R/W
STAT DISC.BAS $R/W
```

nadają zbiorom status Read/Write (Odczyt/Zapis), usuwając wcześniej wprowadzony status Read/Only.

Polecenia:

```
STAT *.COM $SYS
STAT SECRET.BAS $SYS
```

nadają zbiorom status zbiorów systemowych, w wyniku czego zbiory te nie są umieszczane w skorowidzu dysku, wyświetlanym za pomocą DIR oraz stają się niedostępne dla programów kopiujących zbiory. Zbiory te są jednakże nadal dostępne do wszelkich innych celów.

Polecenia:

```
STAT *.COM $DIR
STAT SECRET.BAS $DIR
```

odbierają zbiorom status zbiorów systemowych, w wyniku czego zbiory te są znów uwidaczniane w skorowidzu (DIRectory) dysku.

Program użytkowy FILECOPY umożliwia kopiowanie zbiorów z jednego dysku na drugi przy użyciu jednego napędu dyskowego. Wymagane jest przy tym zmienianie dysków zgodnie z instrukcjami wyświetlanymi na ekranie. W przypadku wprowadzenia nazw zbiorów ze znakami zastępczymi FILECOPY żąda potwierdzenia chęci kopiowania każdego zbioru, odpowiadającego wprowadzonej nazwie. Program wyświetla nazwę każdego kopiowanego zbioru.

FILECOPY *.COM kopiuje wszystkie zbiory typu .COM
FILECOPY PIP.COM kopiuje zbiór PIP.COM

DISCKIT2 spełnia te same funkcje co DISCKIT3, działa jednak nieco wolniej przy kopiowaniu dysków, ponieważ dysponuje mniejszym obszarem pamięci.

Dwa programy użytkowe umożliwiają wymianę zbiorów między dyskami a kasetą magnetofonową. Użytecznie mogą być jednakże wymieniane tylko zbiory typu ASCII, zawierające jawny tekst; wymiana zbiorów o innej zawartości wymaga specjalistycznego doświadczenia.

CLOAD (Cassette LOAD - wprowadzanie z kasety magnetofonowej) wymaga dwóch parametrów: pierwszym jest nazwa zbioru źródłowego (w kasecie), zamknięta w podwójnych cudzysłowach, drugim jest nazwa, pod którą zapisany zostanie zbiór na dysku. Jeżeli druga nazwa jest pominięta, nazwa zbioru dyskowego zostaje taka sama, jak zbioru w kasecie. Jeżeli pominie się nazwę zbioru źródłowego, CLOAD wczyta pierwszy program spotkany na taśmie. Jeżeli pierwszym znakiem zapisanej na taśmie nazwy zbioru jest ! , pominięte zostaną informacje, normalnie wyświetlane podczas czytania kasety.

Przykładowa dyrektywa typu CLOAD to:

CLOAD "MY LETTER" MYLETTER.TXT

CSAVE (Cassette SAVE - przechowywanie w kasecie) może mieć trzy parametry. Pierwszym jest nazwa zbioru źródłowego (na dysku), drugim nazwa zbioru zapisywanego na taśmie magnetofonowej, umieszczona w podwójnych cudzysłowach. Przy pominięciu drugiego parametru zbiór zapisywany jest na taśmie pod taką samą nazwą jak zbiór dyskowy. Jeżeli pierwszym znakiem nazwy zbioru taśmowego jest ! , pomija się informacje normalnie wyświetlane podczas zapisywania na taśmę magnetofonową. Jeśli wprowadzone są obydwie nazwy zbiorów, użyty może być też trzeci parametr, określający szybkość zapisu na taśmę magnetofonową: Ø dla 1000 lub 1 dla 2000 bodów.

Przykładowe dyrektywy typu CASVE to:

CSAVE OUTPUT.TXT "OUTPUT TEXT" 1

CSAVE DATAFILE

SETUP

Program użytkowy o tej nazwie pozwala na zmianę kodów wprowadzanych przez klawiaturę, zmianę parametrów napędu dyskowego i kanału transmisji szeregowej. Umożliwia także inicjalizowanie różnych akcji po wstępnym wprowadzeniu CP/M 2.2. Odmiennie, niż

oddzielne procedury, wchodzące w skład systemu CP/M, które realizują określone działania natychmiast po ich inicjalizacji, SETUP modyfikuje konfigurację sektorów na dysku, wprowadzaną dopiero podczas kolejnej reinicjalizacji systemu. W ten sposób działanie SETUP jest podobne do działania PROFILE.SUB.

Działanie programu określają informacje wyświetlane na ekranie, jeżeli wyświetlane wartości są prawidłowe i nie wymagają modyfikacji, przejście do następnej operacji uzyskuje się odpowiadając Y na pytanie:

Is this correct (Y/N):_

(Czy to w porządku Yes - tak, No - nie)

Program można zakończyć przez wprowadzenie [CONTROL] C. Jeżeli wyczerpany został repertuar podstawowych zmian systemowych, stawiane jest pytanie:

Do you want to update your system disc (Y/N):_

(Czy chcesz zmienić swój dysk systemowy Yes - tak, No - nie)

co daje możliwość zachowania istniejącej konfiguracji sektorów przez wciśnięcie klawisza N . Następnie stawiane jest pytanie:

Do you want to restart CP/M (Y/N):_

(Czy chcesz ponownie zainicjalizować CP/M Yes-tak, No-nie)

Odpowiedź Y pozwala na reinicjalizację systemu i wypróbowanie nowej konfiguracji.

Aby skopiować sektor, zawierający oprogramowanie zależne od konfiguracji należy użyć BOOTGEN lub wprowadzić SETUP z dysku źródłowego, odpowiedzieć Y na KAŻDE pytanie i wprowadzić dysk na który chce się kopiować bezpośrednio przez odpowiedzią Y na pytanie "Do you want to update your system disc (Y/N):_

Znaki o wartości w kodzie ASCII mniejszej niż dziesiętnie 32 mogą być zapisywane w postaci ↑ po której następuje odpowiedni znak ze zbioru @ ,A-Z,[,\,],> ,-

Szczególnej uwagi wymagają zwykle następujące funkcje programu:

** Initial command buffer:

(bufor wstępnych poleceń)

Jakiekolwiek wprowadzone tutaj znaki pojawią się w takiej samej postaci jak zostały napisane jako polecenie po inicjalizacji systemu. Może to być wykorzystane do automatycznego wywo-

łania określonego programu lub funkcji bezpośrednio po inicjalizacji systemu. Treść polecenia należy zakończyć znakami ↑M, opisującymi działanie klawisza [RETURN].

Na przykład aby po inicjalizacji systemu samoczynnie wyświetlić skorowidz dysku, bufor wstępnych poleceń powinien zawierać:

DIR ↑M

a w celu samoczynnego wywołania programu Dr. LOGO do bufora tego należy wpisać:

SUBMIT LOGO2 ↑M

Sign-on string:
(Wstępne informacje)
Jest to komunikat, wyświetlany u góry ekranu po wprowadzeniu systemu CP/M. Zauważ użycie ↑J↑M dla uzyskania przejścia do początku następnej linii. Początek standardowego komunikatu zawiera kody ustawiające odpowiednie kolory ekranu i tryb wyświetlania z 80 znakami w wierszu; początek ten powinien być dokładnie przepisany, jeżeli chce się zachować takie same parametry wyświetlania.

Keyboard translations:
(Tłumaczenie klawiatury)
Pozwala przyporządkować poszczególnym klawiszom nowe wartości kodu ASCII, działając podobnie jak rozkaz KEYDEF w języku BASIC Deklarowanymi parametrami są kody klawiszy i przypisywane im wartości kodu ASCII. Numery klawiszy podane są na rysunku zamieszczonym w prawym górnym rogu pokrywy komputera lub w części rozdziału zatytułowanego "Trochę użytecznych informacji".

Keyboard expansions:
(rozszerzenie klawiatury)
Efektywnie odpowiada rozkazowi KEY w języku BASIC.

Na koniec ...

Programy DISCKIT2, SYSGEN, BOOTGEN, FILECOPY, SETUP, CSAVE i CLOAD zostały opracowane w firmie AMSTRAD i działają wyłącznie w komputerach AMSTRAD z systemem CP/M 2.2. Nie mogą one mieć zastosowania w innych systemach CP/M, chociaż różni wytwórcy mogą dostarczać podobne programy (często o tych samych nazwach), dopasowane do ich sprzętu.

Strona 4 pakietu dysków systemowych zawiera także poniższe programy systemu CP/M 2.2, przeznaczone dla specjalistów; opisy tych programów znaleźć można w "SOFT 159 - A Guide to CP/M" lub innych opracowaniach:

ASM	Asembler mikroprocesora 8080
DDT	Debugger do uruchamiania programów w języku 8080
DUMP	Program listujący zawartość zbioru w kodzie szesnastkowym
ED	Prosty edytor tekstu
LOAD	Zamienia zbiory typu .HEX, tworzone przez ASM na zbiory typu .COM
MOVCPM	Generuje system CP/M 2.2, dostosowany do innego /mniejszego/ obszaru pamięci
SUBMIT	Umożliwia sterowanie systemem rozkazami, umieszczonymi w zbiorze poleceń
XSUB	Steruje wykonywaniem programów przejściowych rozkazami ze zbioru poleceń

Powyższe programy przeznaczone są do użytku w systemie CP/M 2.2 (ze strony 4) i nie należy ich mylić z programami przeznaczonymi dla systemu CP/M Plus (ze Strony 1, 2 i 3).

Cena Tom I/II zł 2000.-

OE
TO
„BIT"